Institut für Logik, Komplexität
und Deduktionssysteme
Universität Karlsruhe
Am Fasanengarten 5
D–76128 Karlsruhe

Datengetriebene Bestimmung von Vokabulareinheiten für koreanische Spracherkennung auf großen Wortschätzen

Diplomarbeit von

Daniel Kiecza

Betreuer:

Prof. Dr. Alex Waibel
Tanja Schultz

angefertigt am

Computer Science Department
Interactive Systems Laboratories
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.

daniel@kiecza.de

October 1999

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Pittsburgh, den 21.10.1999

Daniel Kiecza

Zusammenfassung

Koreanisch zählt zu den agglutinierenden Sprachen. Verwendet man die aus dem Agglutinationsprozeß entstehenden Einheiten – genannt Eojeols – als Wörterbucheinträge für ein Spracherkennungssystem, wächst die Größe des Vokabulars etwa linear mit der Größe des vorliegenden Textmaterials. Außerdem erreicht die out-of-vocabulary (OOV) Rate Größenordnungen, die ein leistungsfähiges Spracherkennungssystem unmöglich machen. Es werden folglich geeignete sub-Eojeol-Einheiten als Wörterbucheinträge benötigt.

Bisherige Arbeiten verwenden hierzu aufwendige Expertensysteme, die die Eojeols in ihre Morphemkomponenten zerlegen. Diese Morphemeinheiten werden dann als Wörterbucheinträge verwendet.

In der vorliegenden Arbeit wird ein neues Verfahren zur Findung geeigneter Einheiten vorgestellt, das ausschließlich *datengetrieben* arbeitet. Ausgangspunkt ist Textmaterial, in dem sämtliche Eojeols in ihre Silbenkomponenten zerlegt sind. Dann werden wiederholt bestimmte Paare von Einheiten zusammengefügt. Die Wahl der zu verbindenen Paare wird dabei so gefällt, daß die akustische Verwechselbarkeit der Einheiten reduziert wird.

Die Erkennungsleistung der resultierenden Systeme wird präsentiert und mit der Erkennungsleistung entsprechender morphembasierter Systeme verglichen. Das beste, auf dem datengetriebenen Ansatz basierende System erreicht eine Eojeol-Fehlerrate von 24.6%. Das entspricht einer Silben-Fehlerrate von 14.5% und einer Phonem-Fehlerrate von 9.9%. Dies vergleicht sich mit einer Eojeol-Fehlerrate von 24.0%, einer Silben-Fehlerrate von 13.0% und einer Phonem-Fehlerrate von 9.4% für das beste morphembasierte System. Die beiden Ansätze bringen vergleichbare Leistung, und man kann folglich auf die aufwendige Konstruktion von Morphemzerlegungssystemen verzichten.

Abstract

Korean is an "agglutinating" language, i.e. verbs and nouns consist of a stem and various appended suffixes which have grammatical function. Using the units that result from the agglutination process, called eojeols, as dictionary entries for a speech recognition system makes the vocabulary size grow linearly in the task size. Furthermore, an extremely high out-of-vocabulary (OOV) rate has to be dealt with which makes the development of a high performance speech recognition system almost impossible. A solution to this problem is to work with sub-eojeol units.

Previous work has used complex expert systems that split eojeol units into their morpheme components. These morpheme components are then used as dictionary units.

This work presents a new *data-driven* approach to determine appropriate dictionary units. The approach starts with a text corpus where the eojeols are split up into their syllable components. Then it repeatedly merges certain pairs of units. The choice of the unit pair to merge is done so as to reduce acoustic confusability of the units.

The recognition performance of the resulting systems is presented and is compared to the performance of morpheme based recognition systems. The best data-driven system we present has an eojeol error rate of 24.6%. This corresponds to a syllable error rate of 14.5% and a phone error rate of 9.9%. The best morpheme based system has an eojeol error rate of 24.0% corresponding to a syllable error rate of 13.0% and a phone error rate of 9.4%. Thus, the two approaches show comparable performance yet we did not need to rely on expert knowledge to generate an *appropriate* set of dictionary units.

Acknowledgements

This work would not have been possible without the help from a number of people.
I would like to thank Professor Alex Waibel for the opportunity to conduct my research at the Interactive Systems Laboratory at Carnegie Mellon University in Pittsburgh.
A big thank you to the supervisor of my thesis, Tanja Schultz, for her advice and guidance during this project. Although our physical separation (me being in Pittsburgh and Tanja being in Karlsruhe) did not make the task of supervision easy we managed to cooperate and communicate effectively and had many very interesting discussions on the telephone and by email.
I also want to mention the many fruitful and inspiring discussions I had with Michael Finke. There, he often suggested solutions when I still tried to understand the problem. These discussions helped me to develop a deeper understanding of the problems involved in this project.
Thank you to Detlef Koll. Sharing his insight was of great pleasure to me and his ideas and suggestions were always very useful.
Then I would like to thank Iain Matthews and Victoria Maclaren for proof-reading and *de-germishifying* this thesis and for helping me to improve my English.
Thank you to Oh-Wook Kwon from the ETRI labs in Seoul, Korea for providing the morpheme segmentation of our text data and also for several interesting discussions via email.
Furthermore, thank you to Monika Woszczyna and Gang-Seong Lee for interesting discussions on aspects related to this work.
Thanks to our assistants at the Interactive Systems Labs in Pittsburgh and Karlsruhe, Debbie Clement, Nadine Reaves and Silke Dannenmaier who always helped to solve administrative issues quickly.
As every computer scientist will understand, I especially want to thank our system administrators Frank Dreilich, Weiyi Yang and Eric Carraux for supporting me in my constant battle with *the hardware*.
Thanks to Sang-Hun Shin, Keal-Chun Cho and Kyung-Kyu Lee for their enthusiasm during collection and validation of the database.
Last but not least, I want to thank my parents who supported me in every possible way during my whole education, especially during the time I spent in Pittsburgh.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Speech is the primary mode of communication among humans. In order to make the interaction between humans and computers more user-friendly, speech must be considered an essential input component.

Over the last two decades, intensive research in the field of automatic speech recognition and a significant increase in available and affordable computing power have led to practical solutions to this problem. Current systems achieve a recognition accuracy between 90% and 100% under certain restricted conditions. Examples of these conditions are a limited vocabulary size in a well defined task domain, a good-quality microphone and a non-noisy environment. One focus of current speech research is to eliminate these limitations.

A great deal of effort is spent developing high performance speech recognition systems for large vocabulary recognition tasks. Each new language that is explored for large vocabulary speech recognition has its own characteristics which can impose difficulties to the development of a high-performance recognition system. This thesis describes our efforts to develop a large vocabulary speech recognition system for the Korean language.

1.2 Problem Definition

For most Western languages, e.g. English, the *word* is an appropriate choice of vocabulary unit. But for Korean, the choice is much more difficult. Korean is an "agglutinating" language. The structure of verbs and nouns is subject to the following rules: A verb consists of a stem and various appended morphemes that have certain grammatical functions, i.e. indicate tempus, modality or social relationship between the communication partners. The syntactic role of a noun is indicated by appending case suffixes to its stem. We call the resulting "meta"-units *eojeols*[1].

Using eojeols as vocabulary units[2] makes the size of the vocabulary grow linearly in the task size. Furthermore, this leads to an extremely high out-of-vocabulary (OOV) rate. This makes the development of a high performance large vocabulary continuous speech recognition (LVCSR) system almost impossible.

Eojeols are built from only about 3600 different syllables[3]. Each such syllable consists of one to four phonemes. Using these syllables as vocabulary units provides a small vocabulary and the OOV rate is 0%. Unfortunately, due to their shortness, two problems arise when this approach is used:

- acoustic confusability of syllable units is very high,
- a standard 3-gram language model has very limited scope.

To overcome the difficulties of using either eojeols or syllables we have to find a more "appropriate" set of vocabulary units that lie inbetween these two extremes.

Recent work on Korean LVCSR makes use of expert systems which are commonly referred to as morpheme tagging systems. These split eojeol units into their morpheme components. As presented in [33], the resulting morpheme units can indeed be used as an appropriate set of vocabulary units for a Korean speech recognition system. Unfortunately, it requires a great deal of effort to develop such a morpheme analyzing system as it involves a lot of *apriori* expert knowledge about the morphological structure of the Korean language.

[1] Pronunciation: [əǰəl].

[2] We use the same set of units in the vocabulary, the dictionary and the language model. These notions are used synonymously throughout this work.

[3] Synonymously called *characters*.

1.3 Contribution of this Work

This work presents a new *data-driven* approach to determine appropriate vocabulary units for Korean LVCSR. The morphemic structure of the Korean language is ignored for the unit determination. Instead, the problem is approached from an acoustically motivated side. Roughly speaking, the data-driven procedure works as follows: In a first pass, each eojeol is split into its character components. Then, repeatedly, syllable pairs are merged in order to reduce acoustic confusability of the phone transitions between units. A vocabulary size of 64k units is used as termination criterion.

The recognition performance of the resulting systems is presented and is compared to the performance of morpheme based recognition systems. This demonstrates whether appropriate vocabulary units for Korean LVCSR may be determined using data-driven methods, without extensive use of *apriori* expert knowledge.

1.4 Organization of the Thesis

This thesis is organized as follows: chapter 2 describes the structural elements of the Korean language which are relevant to this work. In chapter 3 we give a short overview of state-of-the-art speech recognition. The intention is not to give an extensive introduction but to present key concepts and important notations that are needed as a foundation for this thesis. In chapter 4 we describe the speech and text data we used. In addition, we explain how automatic dictionary generation is done and present the baseline structure of our speech recognition system. Chapters 5 and 6 describe the morpheme based approach and our new data-driven approach, respectively. In chapter 7 we present and discuss the recognition performance results of the systems. Finally, chapter 8 draws conclusions and suggests ideas for future work on Korean speech recognition.

Chapter 2

The Korean Language

2.1 Historical Remarks

A common hypothesis is that the Korean language belongs to the Altaic family [18]. Other languages in this family include Manchu, Mongolian and Turkish. But other theories consider Korean an isolated language. It is not clear how long the Korean language has been spoken. Despite geographic proximity of the countries, Korean is quite unlike Chinese and Japanese.

The Chinese writing system and the Chinese culture came to Korea about two thousand years ago. The Chinese influence was very strong and the existing cultural tradition was widely repressed. It is unknown what writing systems existed in Korea before the Chinese era.

The Chinese system was used in Korea until the 15th century. Due to its complexity and difficulty to learn, most people remained illiterate. People from the upper class were able to use the "foreign" Chinese system because of their education, but there is evidence that even they found it difficult. This was due to the significant differences of the two languages, concerning their phoneme inventories as well as their sentence structure.

The Chinese language consists of innumerous single characters which are pronounced monosyllabically and have a morphemic function. They can be pronounced using different tones in order to distinguish homophonic characters. The language is said to be *tonal*. In contrast, Korean is a *syllabic* language, i.e. words are generated by combination of syllables.

After unsuccessful attempts to solve this conflict, King Se-Jong (reign 1419 – 1450) initiated the development of a totally new writing system for the Korean language in the 15th century. In 1420, a Royal institute was established to develop a Korean alphabet system. Thirteen years later these scientists presented their results to King Se-Jong. In 1446 – after this language system had been tested for three years – the King introduced it in a publication known as 훈민정음 (Hun-min Jeong-eum)[1]. A book was published with the same name, containing the background on this system, the reasons for its creation, information on its usage and so forth. King Se-Jong wrote in the book: "Our language is unique, different from that of China therefore we needed an alphabet of our own. Only scholars or people from the upper class get the opportunity to read or write. I felt it is my responsibility to enlighten people as a King so I have made a new 28 letters and I wish everyone can learn this new alphabet easily and use these new letters comfortably in daily life".

The 훈민정음 alphabet was built on a purely phonetic basis and consisted of 28 letters. It has evolved into the modern Korean alphabet, 한글 (han-geul), which has 24 letters. Although Chinese characters are still used today in conjunction with the Korean alphabet, especially in newspapers, high-level communication in Korea is possible without using them.
That is why the Korean language often offers two expressions for a notion, a Chinese one and a pure Korean one. The Chinese characters that are used in Korean evolved over time, especially concerning their pronunciation. Their writing form remained mostly unchanged since the Han dynasty (206 BC – 220 AD).

For over 400 years the new alphabet has been ill-treated by nobilities who still claimed that the Chinese language was the only option. But then the new system became gradually popular among people of literature.

During the 36 years of Japanese invasion in the early 20th century, it was forbidden to read or write han-geul or to speak Korean. During that time the name 한글 was used for the first time to refer to the Korean system. Af-

[1] The right sound to teach people.

ter independence of Japan, 한글 became the official alphabet for the Korean country.

Currently, Korean is natively spoken by about 67 million speakers, 63 million of which live inside Korea. The Korean language is commonly divided into six main dialects. The Seoul dialect is usually referred to as "standard Korean".

2.2 한글 (Han-geul) – The Korean Writing System

The complete Korean alphabet consists of 40 letters. There are 10 basic vowels and 14 basic consonants. In addition, there are 11 compound vowels which are combined among the basic vowels, and 5 double consonants.

Vowel	Symbolizes	Actual Form	Phoneme	Romani-sation
•	Heaven	Not used in isolation	–	–
ㅡ	Earth	ㅡ	/ɨ/	eu
ㅣ	Man	ㅣ	/i/	i

Table 2.1: The three basic vowels.

This section describes the definition of the letters of the Korean alphabet. The following tables show the original form of the letters, their actual form, the respective IPA [1] phoneme symbol for their pronunciation and the respective roman transliteration symbol that will be used throughout this thesis. Romanisation of the Korean language will be discussed in section 2.5.

The symbols of the three most important cosmic elements heaven " • ", earth "ㅡ" and man " ㅣ " are used as a foundation for the vowel system. With these three symbols as building blocks a total of ten basic vowels are created. Two of these are the *earth* and *man* symbols themselves, pronounced as /ɨ/ and /i/, respectively. These three basic symbols are shown in table 2.1.

Old Form	Actual Form	Phoneme	Romanisation
ㅗ	ㅗ	/o/	o
ㅜ	ㅜ	/u/	u
ㅏ	ㅏ	/a/	a
ㅓ	ㅓ	/ə/	eo

Table 2.2: Primary vowels.

Four primary vowels are then created by building combinations of *earth* and *man* with *heaven*. *Earth* is combined with *heaven* above it to form the letter ㅗ, which is pronounced as /o/. Situating *heaven* below *earth* creates the vowel ㅜ, pronounced as /u/. Combining *man* with *heaven* to its left or right results in the letters ㅓ and ㅏ which are pronounced as /a/ and /ə/, respectively. The primary vowels are summarized in table 2.2.

Old Form	Actual Form	Phoneme	Romanisation
ㅛ	ㅛ	/io/	yo
ㅠ	ㅠ	/iu/	yu
ㅑ	ㅑ	/ia/	ya
ㅕ	ㅕ	/iə/	yeo

Table 2.3: Secondary vowels.

Four more vowels – the secondary vowels – are built by combining *earth* and *man* with two *heaven* symbols. Following the same combination concept for the primary vowels, this results in the letters ㅛ, ㅠ, ㅑ and ㅕ with two *heaven* symbols to the top, bottom, right and left, respectively. The pronunciations for these vowels are /io/, /iu/, /ia/ and /iə/. See table 2.3 for a summary.

These ten basic vowels are combined in pairs to form a total of 11 compound vowels. A summary of the original and actual form of all these letters along with their IPA phoneme and roman transliteration symbol is given in table 2.4.

Old Form	Actual Form	Phoneme	Romanisation
ㅐ	ㅐ	/ɛ/	ae
ㅒ	ㅒ	/iɛ/	yae
ㅔ	ㅔ	/e/	e
ㅖ	ㅖ	/ie/	ye
ㅘ	ㅘ	/oa/	wa
ㅙ	ㅙ	/oɛ/	wae
ㅝ	ㅝ	/uə/	weo
ㅚ	ㅚ	/ø/	oe
ㅟ	ㅟ	/y/	wi
ㅞ	ㅞ	/ue/	we
ㅢ	ㅢ	/ɨi/	yi

Table 2.4: Compound vowels.

Five basic symbols were created as a foundation for building consonant letters. These five symbols were shaped after organs in the human articulation system which produce the sounds. Namely these five basic symbols are ㄱ, ㄴ, ㅁ, ㅅ and ㅇ. The tongue is pressed against the **molar teeth** to produce the phoneme /k/ associated with ㄱ. The phoneme /n/ is represented by ㄴ which shows the shape of the **tongue** while this sound is produced. A front view of the **lips** looks like ㅁ while producing an /m/. **Teeth** and tongue are used to produce an /s/ sound. The symbol ㅅ represents this "tooth" sound. And finally, the circle shaped **pharynx** (throat) is used to produce the phoneme /ŋ/. The pharynx is symbolized by ㅇ.

Several more consonants are built by increasing the level of articulation among these five basic ones. This increasing articulation is represented by adding further strokes to a letter.

As stated before, the 한글 letters were developed on a pure phonetic basis.

While the pronunciation of vowel letters is fixed and an IPA sound symbol can be associated easily, we can not provide one simple sound symbol for consonants. This is because the acoustic realisation of a letter is highly dependent on its context. We call this a “variant realisation” or an “allophone”. So, to introduce the consonant letters here we associate them with their roman transliteration symbol only. For a detailed discussion of the phonetic and allophonic characteristics of the consonant letters see section 4.4.1.

The definition form of all 19 consonant symbols is displayed in table 2.5. These symbols are grouped according to the five basic consonant sets defined above. Table 2.6 lists the original symbols again along with their actual form and their roman transliteration symbol.

Producing Organ	Basic Consonants	Extensions	Exceptions
Molar Tooth	ㄱ	ㅋ ㄲ	
Tongue	ㄴ	ㄷ ㅌ ㄸ	ㄹ
Lips	ㅁ	ㅂ ㅍ ㅃ	
Tooth	ㅅ	ㅈ ㅊ ㅆ ㅉ	
Pharynx (Throat)	ㅇ	ㅎ	

Table 2.5: The five basic consonants and their extensions.

2.3 Syllable Construction

The Korean writing system is letter based which makes it fairly easy to learn. The letters are not written sequentially like in the roman writing systems but arranged in syllable complexes, also called *characters*.

The notion of a syllable in Korean is different than in English. A Korean consonant by itself can not form a syllable. This is because a consonant, if not followed by a vowel, can not be released. However, a vowel by itself can form a syllable. In English, a syllable is a sound or a group of sounds accompanied by one of four stresses, whereas in Korean it is a sound or group of sounds which takes up a certain relative space of time like metronome beats.

Old Form	Actual Form	Romani-sation	Old Form	Actual Form	Romani-sation
ㄱ	ㄱ	k	ㅁ	ㅁ	m
ㅋ	ㅋ	kh	ㅂ	ㅂ	p
ㄲ	ㄲ	kk	ㅍ	ㅍ	ph
ㄴ	ㄴ	n	ㅃ	ㅃ	pp
ㄷ	ㄷ	t	ㅅ	ㅅ	s
ㅌ	ㅌ	th	ㅈ	ㅈ	c
ㄸ	ㄸ	tt	ㅊ	ㅊ	ch
ㄹ	ㄹ	r/l	ㅆ	ㅆ	ss
ㅇ	ㅇ	ng	ㅉ	ㅉ	cc
ㅎ	ㅎ	h			

Table 2.6: Original and actual form of the consonant symbols and their romanisation.

A syllable is built according to one of the three structural types CV, CVC and CVCC, where C stands for consonant and V stands for vowel. These letters are arranged in an imaginary square according to the following rules:

- The nine vowels that have the stroke standing in a vertical position, namely ㅏ, ㅓ, ㅐ, ㅑ, ㅕ, ㅒ, ㅔ, ㅖ and ㅣ, are situated to the right side of the syllable's initial consonant, e.g. 나, 더, 비.
- The five horizontal vowels ㅗ, ㅛ, ㅜ, ㅠ and ㅡ are situated below the syllable's initial consonant, e.g. 소, 주, 그.
- The seven diphthongs ㅚ, ㅘ, ㅙ, ㅟ, ㅝ, ㅞ and ㅢ have the syllable's initial consonant on the top of their left sides, e.g. 되, 뒈, 휘.
- The final consonant (CV**C**) or consonant pair (CV**CC**) is put at the bottom edge of the character.
  - With a horizontal vowel it is centered horizontally, e.g. 물, 습, 손, 돐.
  - With a vertical vowel it is located slighty to the right of the horizontal center position, e.g. 각, 강, 섬, 법, 값, 넓.

In the past, the character blocks were commonly lined up top-to-bottom, right-to-left. Nowadays, they are arranged left-to-right, top-to-bottom.

2.4 Phonetics and Phonology

From a phonetic point of view the following types of syllables are present in the Korean language: V, CV, VC and CVC. This differs from the graphical syllable structure presented above in two aspects:

1. A phonetic syllable can have a vowel as its inital sound (V, VC) but a graphical syllable cannot. This phenomen has historic reasons. These syllable types will have the consonant character ㅇ as initial consonant in their graphical form but acoustically this consonant is silent, i.e. it does not have a pronunciation, for example: 이 is graphical CV but phonetical V, 안 is graphical CVC but phonetical VC.

2. The graphical type CVCC does not have a phonetic equivalent. One of the two final consonants is silent unless the second one can be pronounced as initial consonant of the following syllable. This is possible if the following syllable has a vowel as initial phoneme.

A consonant that stands between two vowels is always pronounced as initial consonant of the latter syllable.

Korean letters represent phonemes rather than sounds. Phoneme and sound are sometimes the same, as is the case for the Korean vowels. These phonemes have a fixed acoustic realisation. See section 2.2 for the IPA based description. A detailed description of the proper articulation of these sounds can be found in [18, 39].

It is more complex for consonant characters. Here, most phonemes are realised with different sounds depending on their position in a word and their surrounding sounds. These different realisations are called variant realisations or allophones. The rules for these variants are relatively simple, but to make the description easier, we define the following abbreviations for groups of phonemes beforehand: **C**=consonants, **V**=vowels, **S**=sonorants={ㅁ,ㄴ,ㅇ,ㄹ} and **O**=obstruents={ㄱ,ㄷ,ㅂ,ㅅ,ㅈ,ㅊ,ㅋ,ㅌ,ㅍ,ㄲ,ㄸ,ㅃ,ㅆ,ㅉ}. During the description, we make use of the basic set operator "\". A\B means "set A minus set B".

• The phonemes ㅂ, ㄷ and ㄱ are lenis plosives that are articulated with little tension and without aspiration. They are bilabial, apico-alveolar and dorso-velar, respectively. For each of these phonemes, three allophones exist:

1. Voiceless: [p], [t], [k]
   (a) As word initial sound, e.g. 비, 다 or 그.
   (b) Inside a word after **C\S**, e.g. ㅂ in 학부.
2. Voiced: [b], [d], [g]
   (a) Inside a word between vowels, e.g. ㅂ in 나비.
   (b) Inside a word after **S**, e.g. ㅂ in 공부.
3. Unreleased: [p˺], [t˺], [k˺]
   (a) Word final position, where the word is followed by an articulation break, e.g. ㅂ in 입.
   (b) Inside a word before **C**\{ㅁ,ㄴ,ㄹ,ㅎ}, e.g. ㄱ in 학부.

• The phoneme ㅈ is a lamino-alveolar affricate articulated with little tension and without aspiration. The variants are:

1. Voiceless: [č]
   (a) As word initial sound, e.g. 잘.
   (b) Inside a word after **C\S**, e.g. 극장.
2. Voiced: [ǰ]
   (a) Inside a word between vowels, e.g. 가족.
   (b) Inside a word after **S**, e.g. 남자.
3. Unreleased: [t˺]
   (a) Word final position, where the word is followed by an articulation break, e.g. 빚.
   (b) Inside a word before **C**\{ㅁ,ㄴ,ㅎ,ㅅ,ㅆ}, e.g. 멎다.

• The phonemes ㅍ, ㅌ, ㅋ and ㅊ are strongly aspirated sounds corresponding to ㅂ, ㄷ, ㄱ and ㅈ, respectively. Two positional variants exist:

1. Voiceless: [p^h], [t^h], [k^h], [$\check{c}^h$]
   (a) As word initial sound, e.g. 팔.
   (b) Inside a word after **C** or between vowels, e.g. 남포, 앞에.
2. Unreleased: [p˺], [t˺], [k˺], [t˺]
   (a) Word final position, where the word is followed by an articulation break, e.g. 밭.
   (b) Inside a word before $\mathbf{C}\backslash\{$ㅁ,ㄴ,ㄹ,ㅅ,ㅆ$\}$, e.g. 앞방.

• The phoneme ㅅ is a voiceless apico-alveolar fricative. Before ㅣ and ㅟ it is slightly palatalised. The variants are:

1. Voiceless: [s]
   (a) As word initial sound, e.g. 소.
   (b) Inside a word between vowels, e.g. 무슨.
   (c) Inside a word after **C**, e.g. 항상.
2. Unreleased: [t˺]
   (a) Word final position, where word is followed by an articulation break, e.g. 빗.
   (b) Inside a word before $\mathbf{C}\backslash\{$ㅁ,ㄴ,ㅎ,ㅅ,ㅆ$\}$, e.g. 핫것.

• The phonemes ㅃ, ㄸ, ㄲ, ㅉ and ㅆ are voiceless unaspirated fortis sounds produced with a partially constricted glottis and additional subglottal pressure. They are bilabial, apico-alveolar and dorso-velar stops, lamino-alveolar affricate and apico-alveolar fricative, respectively. Two variants occur:

1. Glottalized: [p'], [t'], [k'], [č'], [s']
   (a) As word initial sound, e.g. 뿌리.
   (b) Inside a word after **C** or between vowels, e.g. 맏딸, 글씨.

2. Unreleased: [k˺], [t˺]

   (a) Syllable final position, followed by an articulation break, only ㄲ and ㅆ appear in this position, e.g. 밖, 있고.

• The phoneme ㅁ is a voiced bilabial nasal. It is pronounced as [m] in all positions.

• The phoneme ㄴ is a voiced apico-alveolar nasal. It is pronounced as [n]. In connection with ㄹ it is subject to assimilation effects. See further below.

• The phoneme ㅇ occurs only in syllable final position and is pronounced as [ŋ].

• The phoneme ㄹ is a liquida phoneme which has the variants [ɾ], a voiced apico-alveolar flap, and [l], a voiced apico-alveolar lateral. These variants appear in the following situations:

1. Flap: [ɾ],

   (a) As word initial sound, e.g. 라디오.
   (b) Inside a word between vowels, e.g. 나라.
   (c) Inside a word before ㅎ, e.g. 말하다.

2. Lateral: [l],

   (a) Word final position, e.g. 달.
   (b) Inside a word before $\mathbf{C}\backslash\{$ㅎ$\}$, e.g. 살다.
   (c) Inside a word after ㄹ, e.g. 물로.

• The phoneme ㅎ is a voiceless glottal fricative. Generally it is pronounced as [h]. In front of ㅣ, ㅟ, ㅑ, ㅕ, ㅛ, ㅠ, ㅖ and ㅒ it is slightly palatalised. In front of ㅜ, ㅚ, ㅟ, ㅘ, ㅝ, ㅙ and ㅞ it sounds like a bilabial [f]. The positional variants are:

1. Word initial position, according to above rules, e.g. 하나.

2. Inside a word, syllable initial position:

   (a) After {ㅁ,ㅇ,ㄹ}, according to above rules.
   (b) After **O**\{ㅅ}, aspiration of preceding consonant, e.g. 잊히다.
   (c) After **V** or ㄴ, ㅎ weakens or even mutes, e.g. 마흔.

3. Inside a word, syllable final position:
   (a) In front of ㄱ, ㄷ and ㅈ, aspiration of that consonant, e.g. 좋고.
   (b) In front of ㅅ and ㄴ, ㅎ mutes; ㅅ is glottalized, ㄴ is lengthened, e.g. 좋소, 좋니.
   (c) In front of **V**, ㅎ weakens or even mutes, e.g. 좋은.

These allophonic rules specify a consonant's pronunciation. In the case of two neighbouring consonants, so called assimilation effects additionally take place which further alter the pronunciation of the involved consonants. These effects can be divided into three groups: 1) neighbouring of two sonorants, 2) neighbouring of an obstruent and a sonorant and 3) neighbouring of two obstruents. These rules are rather complicated. Instead of covering them extensively, here are some examples. A detailed discussion of the assimilation effects can be found in [18, 53].

Two neighbouring sonorants: If ㅁ is followed by ㄹ then this ㄹ will change its pronunciation to [n].

A sonorant neighbours an obstruent: If ㄴ is preceded by ㅂ then this ㅂ will change its pronunciation to [m].

Two neighbouring obstruents: If ㅅ is followed by ㄷ then this ㄷ will change its pronunciation to [s].

2.5 Romanisation

The definition and development of the Unicode character coding system since 1991[2] made it possible to represent the letters and characters of most languages on a computer system. But up to now, most systems have relied on

[2]The Unicode Standard Consortium was founded in 1991. Its goal is to develop "a character coding system designed to support the interchange, processing, and display of the written texts of the diverse languages of the modern world.". Visit http://www.unicode.org for further information.

the 7 or 8-bit ASCII standard which only provides codes for the characters of Western languages. To represent non-Western writing systems, for example the Korean one, it is necesssary to define a mapping of its characters onto (combinations of) ASCII characters. This is called a *transcription system* or a *romanisation.*

Many systems exist for the Korean language, but none is internationally accepted as standard. The *McCune-Reischauer* transcription system is considered the traditional one and despite its inconsistencies and use of diacritic marks, it is the most widely used.[3] Further transcription systems include the ones proposed by the *North Korean* and *South Korean* governments. The *Yale* transcription system has been developed with simplicity and consistency. At present, this system is used by most technical papers in linguistics.

The unix tool *hcode* [72] can convert between several code representations for 한글 characters. We use this tool to convert all our 한글 text data into the hcode-specific transcription system. This system is different from the above mentioned ones, yet it is very similar to the one proposed by the South Korean government.

Tables A.1 and A.2 in the appendix show a summary of these transcription systems.

[3]Libraries have adopted the *McCune-Reischauer* system as their standard. Switching to a new transcription system would be very expensive as for instance all the library references where alphabetical ordering is used would have to be rewritten or updated.

Chapter 3

Speech Recognition

3.1 Motivation

Speech input is more natural than keyboard input as it is less intrusive to the user and he is not bound to a hardware device and does not have to keep his focus on the screen constantly. Therefore, to make the interaction of humans and computers more user-friendly we have to consider speech an essential input component. However, there are tasks, like drawing, where other input modalities are superior to speech.

Applications for speech recognition systems are widespread. For the average user it would be much more convenient to enter text to a *word processing* program via talking than via typing. However, it would, for example, not be feasible to let several employees use speech input for their word processing purposes in a big shared office. Speech recognition systems can be used as part of a *speech-to-speech translation* system which would allow people to communicate using different languages [31]. Large amounts of speech data (broadcast news, interviews, meetings, talk shows etc.) are stored on audio tape and speech recognition could be employed to create a *transcription* of these sources [55,66]. This transcription can then be fed into a database and be indexed for *information retrieval* purposes [25]. *Car* drivers can keep both hands on the steering wheel while they access speech-recognition enabled control instruments like navigation system, radio and cellular phone [59].

Intensive research efforts are still underway to further improve speech recognition technology. Areas of focus include large vocabulary speech recognition [46, 47, 50–52], recognition of spontaneous speech [12] and robust speech recognition in noisy environments [59].

3.2 Overview

Current large vocabulary speech recognition systems are based on the principles of statistical pattern recognition. A front-end acoustic processor (3.1) converts an unknown speech signal S into a sequence of feature vectors $X = x_1, x_2 \ldots, x_m$. The speech recognition system has to find the most probable sequence of words $W = w_1, w_2 \ldots, w_n$ given the parameterised acoustic signal X, i.e. it has to find the word sequence $\bar{W}$ that maximizes $P(W \mid X)$. Using Bayes' rule [10] (3.2) and the fact that P(X) is a constant term in respect to the maximization (3.3), the desired probability can be decomposed as follows:

$$\begin{aligned}
\bar{W} &= \mathrm{argmax}_W \; P(W \mid S) \\
&= \mathrm{argmax}_W \; P(W \mid X) && (3.1) \\
&= \mathrm{argmax}_W \; \frac{P(W) \cdot P(X \mid W)}{P(X)} && (3.2) \\
&= \mathrm{argmax}_W \; (P(W) \cdot P(X \mid W)) && (3.3)
\end{aligned}$$

The term $P(W)$ represents the *apriori* probability of observing the word sequence W, independent of the observed speech signal. For instance, the word sequence "how are you" is apriori much more likely than the sequence "are how you". A *language model* (LM) is used to capture this information.

The second term $P(X \mid W)$ represents the probability of observing the signal X given the word sequence W. This value is determined by an *acoustic model* (AM).

The practical determination of the most likely word sequence $\bar{W}$ requires the solution of a number of difficult problems. The process for finding $\bar{W}$ is called *decoding* and the design of efficient decoders is crucial to the realisation of practical LVCSR systems.

In the description above we assumed that a sentence can be decomposed into word units $W = w_1, w_2 \ldots, w_n$. While this is straightforward for languages like English where the notion of a *word*, as a syntactical spacing unit, is very suitable, the choice of the w_i's is not at all clear in many other languages. Some languages do not use spacing at all and therefore do not have the notion of a *word*. Other languages, like Korean, do have syntactical spacing units, but using these units as w_i's is not suitable for a LVCSR system. This thesis focusses on the determination of appropriate w_i's for a Korean large vocabulary speech recognition system.

In the next four sections we briefly describe each of the components of a speech recognition system. In 3.7 the technique of speaker adaptation is described. It is used to improve recognition performance of speaker independent speech recognition systems. We end the chapter with a short introduction to the Janus Speech Recognition Toolkit in section 3.8.

3.3 Front-End

Computers can only work with discrete data. So, to make computer based speech recognition possible a digital representation of the continuous speech signal is needed. A microphone transforms sound into a continuous electrical signal, then an analog-to-digital converter transfers the continuous electrical signal into discrete time slices and discrete amplitude values. Commonly, this yields 16,000 16-bit samples per second.

It is not usual to use this signal representation directly for the recognition process. First, the amount of data is rather large. Second, the signal still contains a lot of unwanted information like background noise, speaker properties or microphone channel. Thus, several preprocessing steps are used to reduce the amount of data and to extract only the relevant speech information from the signal. The first step in preprocessing is to transform the signal into a spectral representation. Usually, preprocessing systems use a sliding window with a length between 5 ms and 20 ms to extract "frames" of samples from the speech waveform. These frames are commonly extracted every 10 to 20 ms, i.e. the frames usually overlap. Then the discrete Fourier transform [6] is used to transform a frame of time samples into a spectral representation.

Knowledge about the sensitivity of the human ear can be applied to reduce the, commonly over 100, resulting spectral coefficients to about 16 values.

This mapping results in features such as Mel-scale [45].

In order to capture signal changes the feature vector is usually appended with information about neighbour frames. Generally, first and second order differences between the successor and predecessor feature vectors are used. Further features such as signal energy can be added to the vector.

To summarize, the preprocessing is used to obtain a feature vector of around 40 coefficients for about every 10 milliseconds of speech.

Commonly, additional techniques are applied to the resulting feature vector. Among these are:

Mean Normalization The additive stationary parts which are introduced by the channel noise are removed by subtracting the mean of all observation vectors.

Linear Discriminant Analysis (LDA) This is a very efficient technique to reduce the dimension of the feature vector without loss of relevant information. A linear transformation matrix is created based on the classes of the feature vectors. The goal is to build a matrix, with which the feature can be projected into a subspace while keeping or increasing the separability of the classes. The usefulness of linear discriminant analyis in a speech recognition front-end has been widely shown [2, 57, 68].

Vocal Tract Length Normalization (VTLN) One major source of interspeaker variability is the variation in vocal tract shape. Different speakers have different vocal tract lengths. Different vocal tract lengths imply different pitch and formant frequencies. In order to normalize for the length of the vocal tract a maximum likelihood linear or piecewise-linear warping in the frequency axis of the speech signal is performed for each speaker. See [11, 54, 58, 69–71] for more information.

3.4 Language Modeling

The language model captures the probability of a sequence of words $W = w_1, w_2, \ldots, w_n$ and is given by:

$$\begin{aligned} P(W) &= P(w_1) \cdot P(w_2 \mid w_1) \cdot P(w_3 \mid w_1, w_2) \cdots P(w_n \mid w_1, \ldots, w_{n-1}) \\ &= \prod_{i=1}^{n} P(w_i \mid w_1, \ldots, w_{i-1}) \end{aligned}$$

A word's likelihood is calculated based on its *word history* $w_1, \ldots, w_{i-1}$. For a vocabulary of size L there are L^{i-1} different histories, and so to specify the probability completely, L^i values would have to be estimated. This is an extremely large number for practical values of L. As a consequence, the histories must be considered to belong to only a manageable number of different equivalence classes. Let $\pi(\cdot)$ denote a mapping of histories into some number of equivalence classes. Then the probability $P(W)$ may be approximated by

$$P(W) \approx \prod_{i=1}^{n} P(w_i \mid \pi(w_1, \ldots, w_{i-1}))$$

In practice, different equivalence class definitions can be used. The most widely employed approach is the so called *3-gram* model. Here histories are considered equivalent if they end in the same two words. Thus

$$P(W) \approx \prod_{i=1}^{n} P(w_i \mid w_{i-2}, w_{i-1})$$

The estimation of the basic probabilities $P(w_i \mid w_{i-2}, w_{i-1})$ is not trivial. For example, complicated smoothing techniques are used to avoid zero probabilites for 3-grams that do not occur in the text corpus.

For a detailed discussion of LM probability estimation and further LM related issues see [24, 28].

3.5 Acoustic Modeling

Acoustic probabilities can be calculated by representing each word unit with a statistical model. Using a separate model for each word has proven unfeasible for large vocabulary speech recognition systems. It is necessary to obtain several samples of a word from different speakers to train reasonable speaker-independent models for each word, and there are simply too many words to be trained. Also the extension of the recognition vocabulary is very complicated as the training process must be repeated for each new word.

This problem is solved by creating acoustic models for *subword* units. Common choices for subword units are syllables or phones. In fact, phones are most frequently used. As there is only a small and fixed set of phones per language (e.g. about 50 for English) these models can be trained well with a reasonable amount of training data. Adding new words to the recognition vocabulary is then as simple as defining their pronunciation in terms of phones.

Unfortunately, the acoustic realisation of a phone highly depends on the neighbouring phones. This can be addressed by using context-dependent phone models. The first such proposed models took into account the direct predecessor and successor phone and were called triphones [3, 34]. Models that consider the two predecessors and the two successors are called quintphones. A general context-dependent phone model is called polyphone. However, the acoustic features of a phone unit are not constant but change within its boundaries. Consequently, many state-of-the-art recognition systems split each phone into a number of states. These are subsequently called *subphonetic* units. Context-dependent subphonetic units are called *subpolyphone models* [19, 21]. For our systems, we use subquintphone models, meaning we split each phone into three states and model each state with a context width of two.

The key issue in building context-dependent models is to maintain a balance between the desired model complexity and the number of parameters which can be robustly estimated from the available training data [64]. The solution here is to use sufficiently complex models, more than can be trained robustly, and then to cluster similar ones. This can be done bottom-up or top-down. The bottom-up approach is easier to implement, as it works datadependently. The top-down method involves linguistic expert knowledge but it has proven much more powerful in providing suitable models for contexts that do not appear in the training material [35].

The main algorithmic instrument to implement the top-down approach is a phonetic decision tree [5, 8, 38]. At the root of the tree is the set of all polyphones corresponding to a phone. Each node has a binary "question" regarding their left and right contexts. These questions are created using expert knowledge and are designed to capture classes of contextual effects. In general, they are of the form "Does the {next, second-next, previous, before-previous, ...} phone belong to phonetic class x", for example "Is the previous phone a consonant?".

Constructing such a tree is a sequential optimization process which recursively partitions the set of states based on a goodness-of-split criterion. The tree leaves contain the acoustic models. To find the corresponding leaf for a specific polyphone model, the tree is traversed by answering the questions attached to each node, until a leaf node is reached. All models that fall in the same leaf are then represented by the same acoustic model.

Most state-of-the-art systems use hidden Markov models (HMMs) as a statistical representation of the subword units [31, 37, 60]. An HMM is a set of discrete states connected by transitions. Each state can produce an observed feature vector with a certain *output probability*. Each HMM transition from any state i to state j has a static *transition probability*. Mainly left-to-right HMM models are used for speech recognition because speech only "goes forward" and hence no back transitions are needed. Commonly, a phone model is represented by a three state left-to-right HMM as three frames roughly correspond to the average length of a phone unit. Each state has a self-loop, a transition to the next state and sometimes even transitions that skip one or several states. See figure 3.1 for a simple left-to-right hidden Markov model. An HMM state represents a small subspace of the overall feature space. The shape of this subspace is sufficiently complex that it is commonly characterized by a multivariate Gaussian distribution.

Word models can be built by retrieving their subword model representation from a pronunciation dictionary and connecting the HMMs of those subword units.

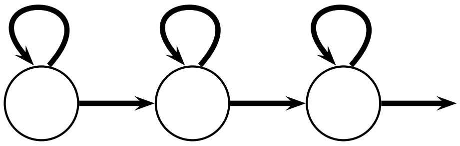

Figure 3.1: Example of a left-to-right hidden Markov model.

There are several reasons why HMMs are the most widely used approach to calculating $P(X \mid W)$. Speech production can be modelled as a stochastic process. A phone is pronounced differently by different speakers; even the same speaker will pronounce a phone differently at different times. The statistical nature of HMMs along with their ability to model temporal processes make them very suitable for this task. Furthermore, HMMs have been researched for a long time and efficient algorithms exist for training and evaluation of these models.

The theory and the practical use of HMMs in speech recognition is covered very well in the literature. The first theoretic work on HMMs was done by Baum [4] in 1972. A detailed introduction to the theory of HMMs can be found in [20, 42]. A general overview of the speech recognition problem may be found in [43, 44, 61]. They also have very extensive bibliography sections. A good introduction into statistical speech recognition is given in [23]. A very good overview of large vocabulary speech recognition systems is given in [65], and [56] contains a broad spectrum of selected speech recognition related papers.

3.6 Decoding

Finding the most likely sequence of words $\bar{W}$ is a complicated search problem. As with all search problems, there are two main approaches: *depth-first* and *breadth-first*. In depth-first designs, the most promising hypothesis is pursued until the end of the utterance is reached. Depth-first decoders are commonly called *stack-decoders* or *A*-decoders* [22, 26, 40, 41]. In breadth-first designs, all hypotheses are pursued in parallel. Breadth first decoders are often referred to as *Viterbi decoders* [62]. Sophisticated pruning techniques are employed to reduce the very complex search space. A process called *beam search* is typically used for this purpose [17].

3.7 Speaker Adaptation

Speaker adaptation is a technique used to adjust the acoustic models of a speaker independent recognition system to a specific speaker in order to increase recognition performance. Its general usefulness has been widely proven [36]. Adaptation is divided into two classes: supervised and unsupervised. In the case of supervised adaptation the real textual transcription of an utterance is known. In the case of unsupervised adaptation the transcription is unknown.

We used speaker adaptation for two purposes: Firstly, supervised adaptation is employed to improve the "labeling"[1] of the training utterances. This technique is also referred to as *label-boosting*. Secondly, we use speaker adaptation in the recognition stage to improve performance. As the actual transcription of a test utterance is not known beforehand, the following strategy is used: in a first recognition pass the best hypothesis is calculated. This hypothesis is assumed to be the actual transcription of the test utterance. Using this transcription, supervised speaker adaptation can be performed. Finally, a second recognition pass is done with the speaker adapted acoustic models.

3.8 The Janus Speech Recognition Toolkit

The Janus Recognition Toolkit (JRTk) [12, 31, 32, 62, 63, 67, 68] is a speech recognition system for research and development. It has been developed at the Interactive Systems Labs jointly at Carnegie Mellon University and the University of Karlsruhe. This toolkit is embedded into a Tcl/Tk interpreter

[1] For time efficiency reasons, generally the mapping of speech frames to acoustic models of a training utterance is not done dynamically during training but a fixed mapping is calculated before the training stage. This process is usually called "labeling".

which allows a user to create a complex recognition system easily while still retaining control over every single variable of the recognition backend. Main benefits from this scripting language interface are a maximum flexibility and a very good extensibility.

The JRTk allows for a large variety of recognition system architectures. It can handle semi-continuous to fully-continuous Gaussian mixture observation models. The Gaussians can be modeled with radial, diagonal or full covariance. Neural nets can also be used to do the acoustic modeling [14, 15].

A common architecture of a JRTk recognition system uses three sub-polyphone models per phone. These sub-polyphones are clustered to around 3,000 sub-allophones [13]. To model each of them a multivariate Gaussian with 16 to 48 diagonal covariance components is used.

JRTk uses a multi-pass search strategy for the decoding of an utterance. The two first passes are Viterbi beam search based and produce a word hypothesis graph, also referred to as word lattice. The third pass, which is called lattice rescoring, can make use of higher order language models to extract hypotheses from that word lattice. See [13, 61, 62] for further details.

Chapter 4

System Description

4.1 Speaker Database

We use the Korean portion of the *GlobalPhone* database [46–52] for the development and evaluation of our systems. This section consists of 20 hours of speech data spoken by 100 native Korean speakers. Each speaker read several articles from a Korean national newspaper. The articles were chosen from the areas: national politics, international politics and economy. The speech data was recorded in stereo at a sampling rate of 48kHz using a close-talking microphone connected to a DAT-recorder. After the sound data was transferred from the DAT-recorder to a hard disc, it was downsampled to 16kHz, 16bit.

	Training	Test	Test subset
Speakers	80	10	10 (same)
Utterances	6,350	798	84
Vocabulary (eojeols)	41,876	7,338	923
OOV rate (OOV words)	–	40.07% (4535)	41.43% (440)
Total utterances	7,148		
Total vocabulary (eojeols)	45,983		

Table 4.1: Summary of acoustic database.

Eighty of the speakers were used for training the acoustic models. They spoke a total of 6,350 utterances with a vocabulary size of 41,876 eojeols.

Ten speakers were chosen as test speakers. The remaining ten speakers are kept as a further cross-validation set. The test speakers spoke a total of 798 utterances with a vocabulary size of 7,338 eojeols and an OOV rate of 40.07%.
A subset of 84 uniformally selected utterances from the ten test speakers was used to carry out our experiments. The vocabulary size of this test set is 923 at an OOV rate of 41.43%. See table 4.1 for an overview of the database.

4.2 Language Model Data

To overcome the sparse data problem in language model generation, we collected a large corpus of text data from the internet. We retrieved the online articles of the Korean newspaper *Chosunilbo* [72] from October 1995 to August 1998. A text preprocessing script cleaned the text data by removing all HTML-related code. Numbers were mapped onto their textual transcription. Acronyms were replaced by mapping each letter onto a transcription of its pronunciation. See table 4.2 for the mapping table. Table B.1 in the appendix shows some examples of acronyms along with their transcription. We mapped acronyms that are pronounced as a word rather than a sequence of letters (e.g. FIFA, OPEC, RAM, UEFA) onto a transcription of that pronunciation. Table B.2 in the appendix summarizes mappings that translate units like *mm*, *GB*, *%* or *kbps* into an appropriate textual description. The text processing script finally dropped all sentences which still contained non-한글 characters (such as Chinese) as our speech recognition system is based on a pure 한글 database.

In the following description, we will refer to the transcription text data of the training speakers as *Train*. The transcription text data of the test speaker transcription will be called *Test* and the subset thereof which is actually used for system evaluation (84 utterances) will be called *Test-Utts*. Refer to table 4.1 for an overview.

The text corpus that was retrieved from the internet plus the corpus *Train* will be referred to as *Chosun+Train*. This corpus has a total size of 14,770,769 eojeols and consists of 1,494,509 different eojeols. In terms of characters the total corpus size is 43,332,100 and the vocabulary size is 3,583. To ensure a time efficient evaluation of our unit determination process and the resulting

Roman letter	한글 and transcription	Roman letter	한글 and transcription
A	에이 (e-i)	N	엔 (en)
B	비 (pi)	O	오 (o)
C	씨 (ssi)	P	피 (phi)
D	디 (ti)	Q	큐 (khyu)
E	이 (i)	R	알 (al)
F	에프 (e-pheu)	S	에스 (e-seu)
G	지 (ci)	T	티 (thi)
H	에이치 (e-i-chi)	U	유 (yu)
I	아이 (a-i)	V	브이 (peu-i)
J	제이 (ce-i)	W	더블유 (teo-peul-yu)
K	케이 (khe-i)	X	엑스 (ek-seu)
L	엘 (el)	Y	와이 (wa-i)
M	엠 (em)	Z	제트 (ce-theu)

Table 4.2: Mapping table for acronyms.

systems we decided to use only about 15% of the large corpus plus the corpus *Train*. This data is referred to as *PartChosun+Train*. The large corpus is only used for some selected experiments. *PartChosun+Train* has a total size of 2,354,072 eojeols. It consists of 417,648 different eojeols. In terms of characters the corpus size is 6,854,294 and the vocabulary size is 3,002. Table 4.3 summarizes the language model corpora information.

	Chosun+ Train	PartChosun+ Train	Train
Number of eojeols	14,770,769	2,354,072	92,378
Eojeol vocabulary size	1,494,509	417,648	41,876
Number of characters	43,332,100	6,854,294	303,203
Character vocabulary size	3,583	3,002	1,963

Table 4.3: Summary of language model corpora.

The *self-coverage* of a text corpus is a one-dimensional function $f(x)$ where x runs from 0 to the size of the vocabulary ($vocabSize$). $f(x)$ lies between 0 and 1 and indicates what fraction of the words of a text is covered when those x words which occur the most often in that corpus are known. Figure 4.1 shows the eojeol-based self-coverage for the corpora *Train*, *PartChosun+Train* and *Chosun+Train*.

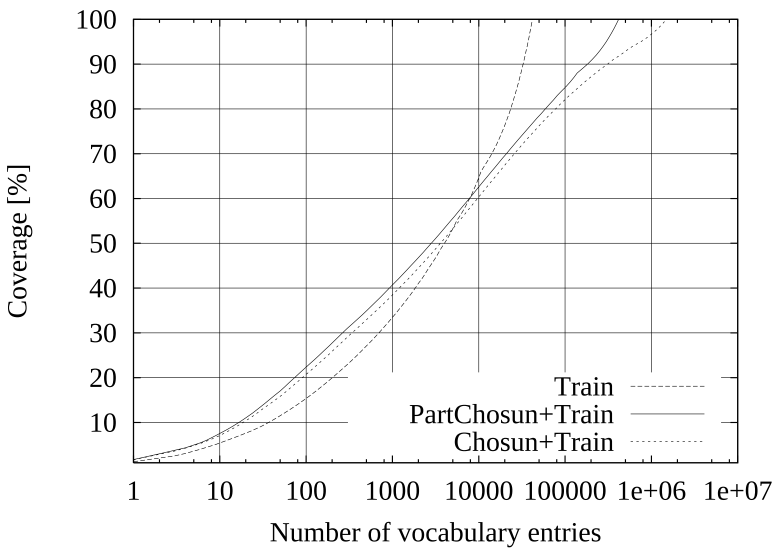

Figure 4.1: Self coverage of language model corpora.

The *cross-coverage* indicates the percentage of *Test* text words covered when the x words are known which occur the most often in the language model corpus. Again, $f(x)$ lies between 0 and 1 but it is not necessarily $f(vocabSize) = 1$. This is because OOV words in the test text cannot be covered as they do not appear in the language model corpus. Figure 4.2 shows the eojeol-based cross-coverage of *Test* using the corpora *Train*, *PartChosun+Train* and *Chosun+Train*, respectively.
Figure D.1 shows the eojeol-based cross-coverage of *Test-Utts* using the same corpora as above. Figures 4.2 and D.1 illustrate the problems using eojeol-based Korean speech recognition. The complete vocabulary of *Train* (41,876 words) covers around 60% (59%) of *Test* (*Test-Utts*) which corresponds to a OOV rate as high as 40.07% (41.43%). *PartChosun+Train* covers up to 80.21% (79.38%) of *Test* (*Test-Utts*). In terms of OOV this is a rate of 19.79% (20.62%). When the complete vocabulary of over 1.2 million eojeols of *Chosun+Train* is used 87.67% (88.80%) of *Test* (*Test-Utts*) are covered. This corresponds to an OOV rate of 12.33% (11.20%). Even this OOV rate is

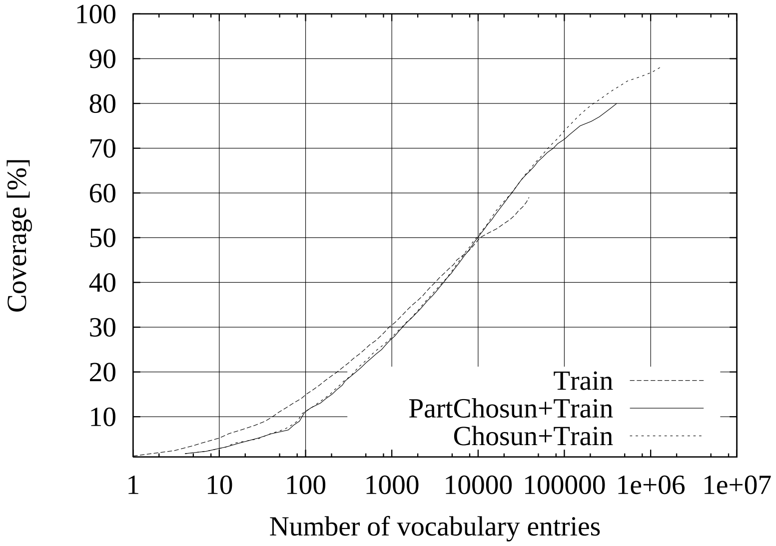

Figure 4.2: Cross coverage of *Test* with different language model corpora.

still unacceptably high for an LVCSR system. The Janus speech recognition toolkit allows for a maximum recognition vocabulary of 64k words. In this case, neither of the mentioned language model corpora cover more than 70% of *Test* or *Test-Utts*, an OOV rate of 30% and above. See table 4.4 for a summary of the OOV information.

	Test	*Test-Utts*
Train	40.07% (4535)	41.43% (440)
PartChosun+Train	19.79% (2239)	20.62% (219)
Chosun+Train	11.20% (1268)	12.33% (131)

Table 4.4: Summary of OOV rates and, in parantheses, OOV words.

Assuming that a speech recognition toolkit did not have any limit for the recognition vocabulary, then gathering as much text data as possible, as the above figures suggest, could be a means of lowering the OOV rate. Unfortunately, this would not work as the vocabulary grows almost linearly in the text data size as figure 4.3 shows. Thus, satisfactory eojeol-based text coverage can not be reached in practice.

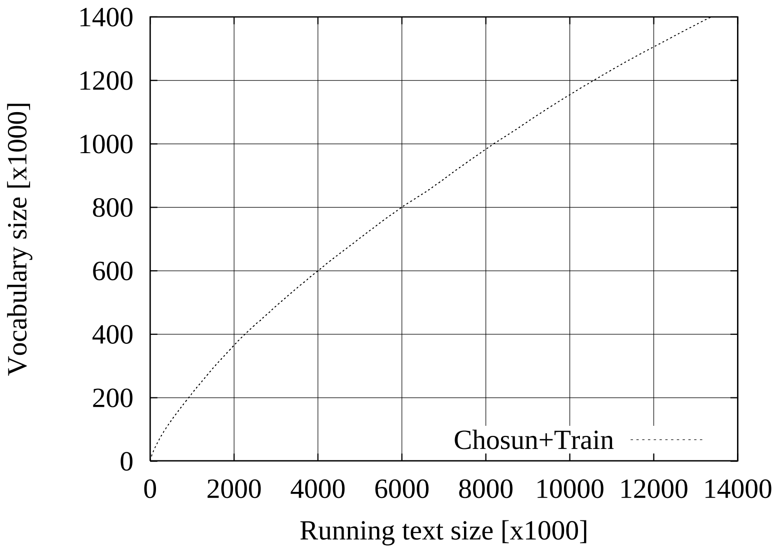

Figure 4.3: Vocabulary growth in corpus *Chosun+Train*.

4.3 Pronunciation Generation

An essential component of an HMM based speech recognizer is the pronunciation dictionary. This dictionary defines a sequence of HMM phone models for each vocabulary unit. For many languages, such a dictionary can only be built by manually editing, which is a strenuous task. Fortunately, this is not the case for Korean. A complete collection of the rules for the allophonic variants of consonant phonemes, as described in 2.4, and the phonological rules, like assimilation, reinforcement and weakening for neighbouring consonants, also in 2.4, makes it possible to generate the pronunciation dictionary automatically.

The basic strategy for automatic dictionary generation is to take a corpus of text data and then generate the pronunciation for each word found by applying the above set of rules.

As an example, we consider the word "cheon-ku-paek-o-sip-nyeon"[1]. Applying the set of rules returns the phone sequence [čəngubɛgosimniən]. As we can see, the allophonic rules are processed correctly, for example the letter *k* of the syllable *paek* is mapped on the voiced sound [g] because of its intervocalic position. The neighbouring letters *p* and *n* between the characters *sip* and *nyeon* show an example for the correct treatment of the phonological rules. The two letters are mapped onto the phones [m] and [n], respectively.

Handling phonological changes inside a vocabulary unit is straightforward – simply apply the defined set of rules. However, phonological changes can also occur at unit boundaries. To handle these cases, we extract the last character of the preceding unit and the first character of the succeeding unit and connect them respectively to the beginning and end of the current unit. Now the set of rules can be easily applied, phonological changes happen within the newly created meta-unit. After the corresponding sequence of phones is created, the phones that belong to the two added characters are removed. As a result, we obtain the pronunciation of the current unit in the given context.

Consider the sentence "u-ri cip ro-cheon-ne mal-i-ci". To create a pronunciation for "ro-cheon-ne" in this context, we first build the meta unit "cip-ro-cheon-ne-mal" by connecting the two neighbouring characters. Applying the allophonic and phonological rules results in the phone sequence [ǰimnočənnemal]. Removing those phones that belong to the neighbouring syllables which here are [ǰim] and [mal] leaves us [nočənne] as the pronunciation. Of course this procedure might return different pronunciations for a specific unit depending on the context. These are handled as pronunciation variants in the recognizer's dictionary.

4.4 HMM Recognizer Structure

4.4.1 Phone Set

Based on the results presented in section 2.4 a total of 41 phones, 10 vowels, 8 diphthongs and 23 consonants were defined. A list of these phone models can be found in the appendix. The vowel models are listed in table C.1, the consonants are presented in table C.2. The two tables show for each model

[1]Korean for: *year 1950.*

its IPA representation, the Janus phone model name and an example English word which contains (an approximation of) the specific phone.

We chose not to represent the diphthongs [oɛ], [iɛ] and [ue] as separate acoustic models but rather to break them up into the two respective monophthong models. These three diphthongs are not well enough represented in the acoustic training material to ensure a reliable model parameter estimation. For the same reason we joined the consonants [p] and [p^h], [t] and [t^h] and also [k] and [k^h].

In addition to the phone models we have one silence model (SIL) and one acoustic model that represents human non-speech articulatory noises (+hGH).

The phone models and the noise model are represented as a three-state, left-to-right HMM (see figure 3.1). For the silence model one four-state left-to-right HMM is used. The output probability of each of these HMMs is modeled with a mixture of 16 diagonal 24-dimensional Gaussians.

4.4.2 Speech Preprocessing

The general problem of turning speech data into a form that can be processed by an automatic speech recognition system was described in section 3.3. Based on that section we describe here the final speech feature vector and which preprocessing steps were used.

A window of size 20ms was shifted over the discretized speech data with an offset of 10ms. For each window 13 Mel-frequency cepstral coefficients were calculated. Mean substraction was applied to remove the stationary characteristics of the recording channel. Then a composite 43 dimensional feature vector was generated from the 13 Mel-coefficients, their first and second order derivatives and zero crossing plus logarithmic signal energy and its first and second order derivative. The final 24 dimensional feature vector is computed by an LDA transformation of this 43 dimensional feature vector. Vocal tract length normalization is applied to minimize speaker differences. See section 3.3 for references to literature covering these techniques.

4.4.3 Context Dependent Phone Modeling

All context-dependent systems presented in this work consist of 3000 sub-quintphone models. Crossword models were used to capture contextual effects between words. For algorithmic reasons, maximum context width across words is one. The decision tree used for the context dependent models was generated using a set of 63 phone sets. These sets are listed in table C.3 in the appendix. Further information on context dependent modeling can be found in section 3.5.

4.4.4 Model Initialization

An initial context-independent Korean recognition system was trained using the labels generated by a multi-lingual speech recognition system. Among the languages of this system are German, English, Japanese and Spanish. The Korean phone models were initialized with heuristically chosen close equivalents in the multi-lingual system.

Chapter 5

Morpheme Based Recognition

5.1 Motivation

As described in chapter 1 an eojeol unit consists of a combination of morpheme components. For large vocabulary speech recognition, this structure is a severe problem. Consider as an example the noun 학교 which means *school*. It can occur in a text corpus in many different ways. In the subject form the word would appear as 학교가, in the genitive form it would appear as 학교의, in the direct object form it would appear as 학교를 and so forth. Each time a grammatical suffix is added to the noun's stem to indicate its case. In fact, suffix appending to nouns is not only used to indicate their case, but also to indicate the number or prepositional function. For verbs, among the functions of a suffix are indication of tempus, modality and social relationship between the communication partners. It is easy to see that one or more forms of a noun may appear in the text corpus that is used to build the recognition system. Also, when presented with new text to recognize, we can expect to encounter a previously unseen form of that noun. This is also the case with verbs.

A straightforward approach to Korean speech recognition is to use morpheme units instead of eojeols as the base vocabulary. The OOV rate of morpheme based systems is below 5% which makes these units much more suitable. As described in [29] the morpheme units can be used effectively as dictionary units for Korean LVCSR.

5.2 Determination of Units

It is very time consuming and unfeasible to edit very large text corpora by hand in order to split up each eojeol into its morpheme components. Automatic morpheme analyzing systems must be built which break up an eojeol into its morpheme components. These systems are also referred to as part-of-speech (POS) tagging systems and generally use a knowledge based approach. Each resulting morpheme unit receives a part-of-speech tag which indicates its grammatical function.

Unfortunately, the development of a morpheme analyzer involves a lot of human expert knowledge about the morphological structure of the Korean language. And as is always the case with rule-based approaches, these systems include errors and omissions. But generally, these errors are tolerable as they are consistent throughout the underlying text data.

The morpheme segmentation was provided by Oh-Wook Kwon from the ETRI lab in Seoul, Korea.

5.3 Speech Recognition Systems

We evaluated two morpheme unit based approaches. The first one – called *Morph* – uses the pure morpheme components that are generated by the morpheme analyzing system as vocabulary units. The second system – called *MorphTag* – makes use of the POS tag which is associated with each morpheme unit. In this case, the combination of each morpheme with its POS-Tag, `morpheme+<POS-tag>` is used as vocabulary units. The idea is to give the recognition system a means, via the language model, to determine which POS-tags can follow each other, and thus increase the score for hypotheses that consist of valid morpheme sequences.

As the system *Morph* does not use POS information but only the plain morpheme units it can be compared directly to the units generated by our data-driven approach. It allows us to compare whether the morphemes are a better choice than an automatically generated set of units. The system *MorphTag* will demonstrate whether the use of the additional POS information helps to increase the performance of the morpheme based recognition system.

For the baseline comparison of the morpheme approach and the data-driven approach, we used the corpus *PartChosun+Train* instead of *Chosun+Train* for efficiency reasons. In further selected experiments we also evaluated the morpheme systems on the complete corpus *Chosun+Train*. These systems will be referred to as *MorphWhole* and *MorphTagWhole*.

In the rest of this section we will describe the characteristics of the morpheme based systems. Performance analysis and discussion of the two unit determination approaches is presented in section 7.3.

	Chosun+ Train	*PartChosun+ Train*	*Train*	*Test*
Number of morphemes	26,781,814	4,249,723	185,608	21,973
Morph vocabulary size	349,264	116,529	17,408	4,981
MorphTag vocabulary size	411,697	134,081	20,038	5,621

Table 5.1: Summary of morpheme based language model corpora.

The corpus *Train* has a size of 185,608 morphemes, its *Morph*-based vocabulary size is 17,408 units, for *MorphTag* it is 20,038. The *Test* corpus contains 21,973 morphemes, 4,981 different ones and 5,621 *MorphTag* units. The total size of *Chosun+Train* is 26,781,814 morphemes and it consists of 349,264 morphemes or 411,697 *MorphTag* units. *PartChosun+Train* contains 4,249,723 morphemes at a morpheme vocabulary size of 116,529 and a *MorphTag* vocabulary of 134,081 units. Of course, the vocabulary size of *MorphTag* is bigger than *Morph*. This is because a *Morph* unit can have more than one morphemic function and so can be associated with several POS tags, resulting in more than one *MorphTag* unit. Table 5.1 summarizes this information.

The *Morph*-based self-coverage for the corpora *Train*, *PartChosun+Train* and *Chosun+Train* is displayed in figure 5.1. Figure 5.2 shows the self-coverage for the corpora *Train*, *PartChosun+Train* and *Chosun+Train* in the *MorphTag*-based setup. The *Morph* and *MorphTag*-based cross-coverage of *Test* using the corpora *Train*, *PartChosun+Train* and *Chosun+Train*, respectively is shown in figure 5.3 and 5.4, respectively.

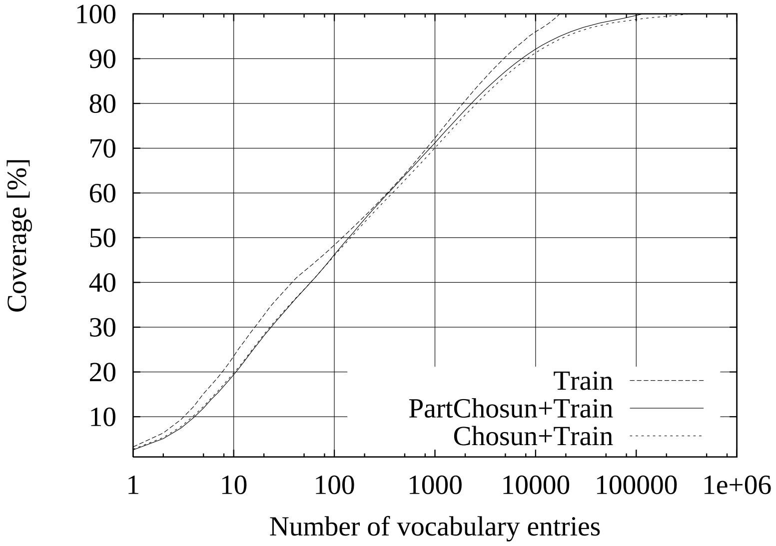

Figure 5.1: Self coverage of *Morph*-based language model corpora.

The analog information for the corpus *Test-Utts* using the same corpora as above is shown in figure D.2 and D.3, respectively. These figures do not differ significantly from the cross coverage figures for *Test* and, as a consequence, are presented in the appendix. We see from these diagrams that the coverage for morpheme units is more consistent than for eojeols; the self coverage and the cross coverage diagrams are almost identical. For the eojeol units the cross coverage is about five percent lower than the corresponding self coverage as can be seen from the respective diagrams in section 4.2.

The *Morph* and *MorphTag* based vocabulary growth is drawn in figure 5.5. We see that the vocabulary grows significantly slower than on an eojeol based corpus. Cross coverage is far above 90% for both *Morph* and *MorphTag* when using a 64k vocabulary. To be precise, the *Morph* system has an OOV rate of 2.37% with a vocabulary of 64k, for *MorphTag* the rate is at 2.78%. These figures are calculated on the corpus *PartChosun+Train*. See table 5.2 for a summary of the OOV information. To summarize, morpheme units are much more suitable for a Korean recognition system than eojeol units.

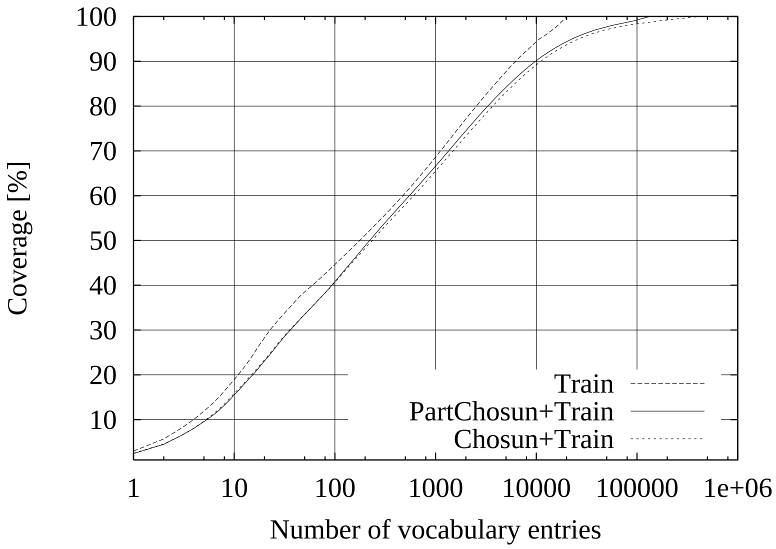

Figure 5.2: Self coverage of *MorphTag*-based language model corpora.

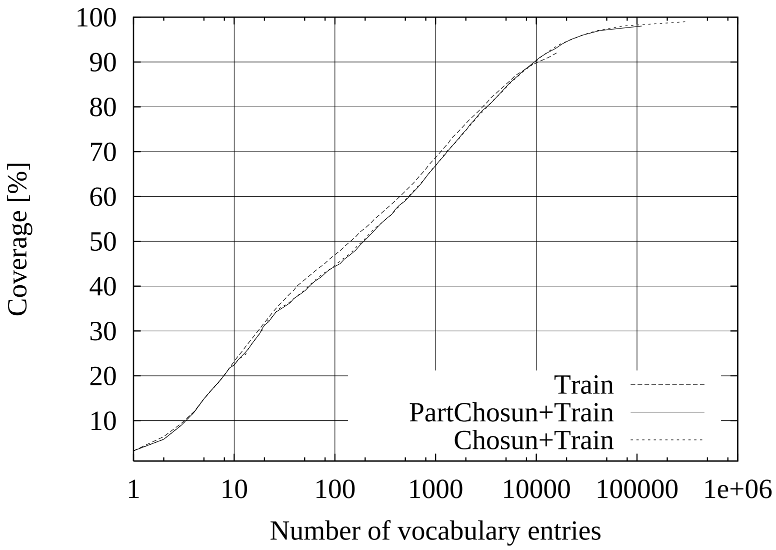

Figure 5.3: *Morph*-based cross coverage of *Test* with different language model corpora.

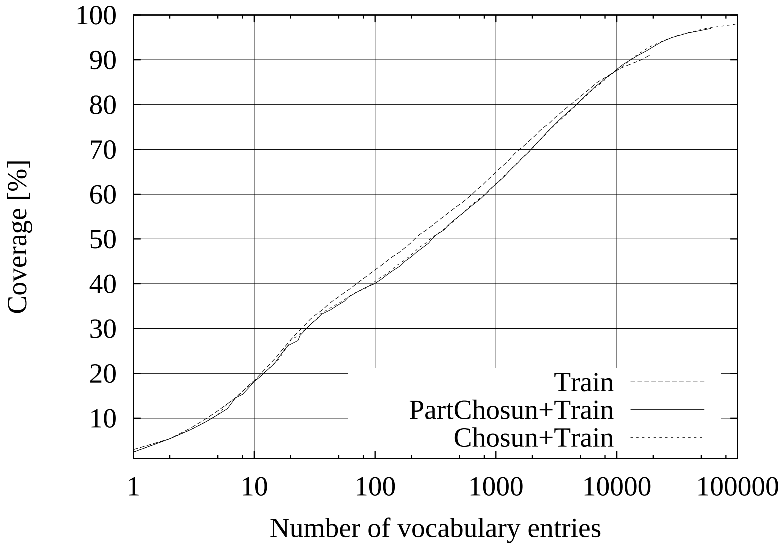

Figure 5.4: *MorphTag*-based cross coverage of *Test* with different language model corpora.

	Test	*Test-Utts*
Morph		
Train	7.44% (1631)	8.40% (176)
PartChosun+Train	1.95% (427)	2.20% (46)
Chosun+Train	0.93% (203)	0.96% (20)
MorphTag		
Train	8.37% (1837)	9.50% (199)
PartChosun+Train	2.22% (486)	2.67% (56)
Chosun+Train	1.12% (245)	1.19% (25)

Table 5.2: Summary of OOV rates and, in parantheses, OOV words.

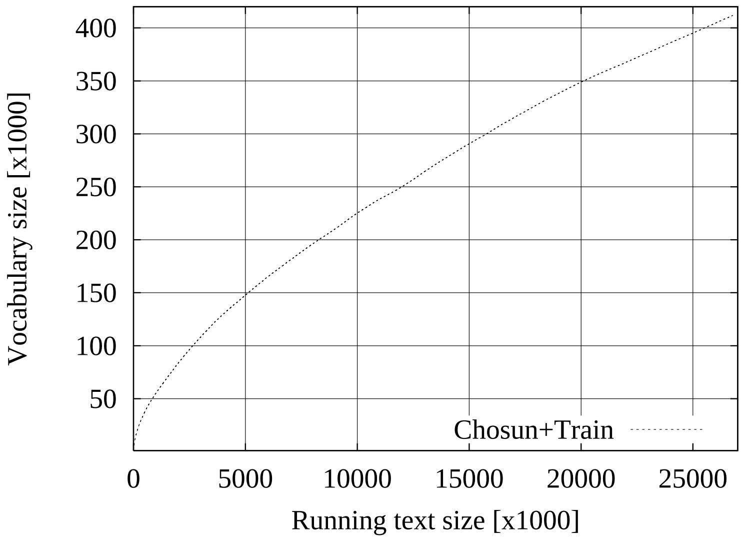

Figure 5.5: Morpheme based vocabulary growth in corpus *Chosun+Train*.

Chapter 6

Data-Driven Unit Determination

6.1 Motivation

Our goal is to generate vocabulary units which on one hand are longer than characters, reducing acoustic confusability, and increasing the range of the 3-gram language model. On the other hand, the units must be shorter than eojeols to retain the OOV rate at a manageably low level.

In preliminary experiments we developed a speech recognition system based on characters as vocabulary units. We presented results of character based recognition systems in [27]. An analysis of the recognition results revealed one dominant type of recognition error: The confusion of pairs of character pairs which share the same sequence of phones from the center vowel of the left character to the center vowel of the right character. We will refer to this phone sequence as *vowel-to-vowel transition*, or simply *transition*. Consider the following examples: The two character pairs *sin eop* and *si neo* share the same vowel-to-vowel transition: *I N EO*. The transition of *math neun* and *manh eun* is *A N EU*, for *ceon hyeo* and *cheo nyeo* it is *EO N iEO*. It is such pairs that are often confused during recognition.

A good language model would certainly help to avoid this type of confusion error. But from an eojeol perspective, the 3-gram language model based on character units has a very limited scope. And in fact, experiments show that this language model does not help to avoid these errors.

The data-driven approach taken here tries to address this problem. Decreasing the average number of confusable vowel-to-vowel transitions between vocabulary units should result in a significant increase in recognition performance. The basic idea of this approach is to repeatedly merge specific unit pairs and thus to *lock* the vowel-to-vowel transition between these pairs *inside* the newly created units. This reduces the number of unit pairs that share a particular vowel-to-vowel transition. In addition the increased average length of vocabulary units should also help the 3-gram language model by increasing its scope.

6.2 Determination of Units

6.2.1 Preprocessing

For the further explanation we consider the sentence *o-neul han-kuk-e ka-yo*[1] as an example.

First, we retrieve all character pairs that appear in the text corpus. The example sentence contains the character pairs *o neul*, *neul han*, *han kuk*, *kuk e*, *e ka* and *ka yo*. Then, for each character pair its vowel-to-vowel transition is generated, for example *han kuk* $\rightarrow$ A N G U. Pronunciation generation is done automatically as described in section 4.3.

For each such character pair we count how often it appears in the text corpus. A hash table with two keys is used to store this information. The first key in the table is the phone sequence of a vowel-to-vowel transition, the second key is the character pair that produces this transition. The value that is associated with these two keys is the occurence count of this character pair. So, if for instance the character pair *han kuk* occured 48,952 times in the text corpus, the hash table would have a key *("A N G U", "han kuk")* that would be associated with the value 48,952. For each first key in the hash table we also store the total sum of occurrences. This value is stored using "total" as second key. Thus, the total number of times that the pronunciation transition *A N G U* appears in the text material could be accessed as

[1] Today I go to Korea.

table.value("A N G U", "total")[2].

For illustration purposes we list some example entries of the resulting hash table:

$$\begin{array}{ll}
\textit{table.value(``A N G U'', ``total'')} & = 83{,}021 \\
\textit{table.value(``A N G U'', ``han kuk'')} & = 48{,}952 \\
\textit{table.value(``A N G U'', ``san ku'')} & = 314 \\
\textit{table.value(``A N G U'', ``pan kun'')} & = 239 \\
\quad\vdots & \\
\textit{table.value(``A N G U'', ``than kuk'')} & = 1 \\
\textit{table.value(``iEO N Ph iEO'', ``total'')} & = 1{,}008 \\
\textit{table.value(``iEO N Ph iEO'', ``yeon phyeong'')} & = 533 \\
\textit{table.value(``iEO N Ph iEO'', ``myeon phyeong'')} & = 127 \\
\quad\vdots & \\
\textit{table.value(``iEO N Ph iEO'', ``myeon phyeo'')} & = 1 \\
\quad\vdots &
\end{array}$$

6.2.2 Unit Merging

The unit merging process is controlled by the following iterative procedure:

1. Choose the vowel-to-vowel transition t that has the highest occurrence count, i.e. the highest *total* value in the hash table:
   $t = \text{argmax}_{key}\ \{\text{table.value}(key, \text{``total''})\}$
2. Select a certain set P of one or more character pairs which produce this transition:
   $P \subseteq \{(p_{i1}, p_{i2}) \mid \text{table.isKey}(t, (p_{i1}, p_{i2}))\}$
3. Merge all character pairs in set P in the text corpus:
   $\forall\ (p_{i1}, p_{i2}) \in P : \text{mergeInCorpus}(p_{i1}, p_{i2})$

[2]We use pseudo code to describe operations on the table data structure: *table.value(a,b)* returns the value that is associated with the key pair (a, b), *table.isKey(a,b)* returns a boolean value indicating whether the key pair (a, b) is in the table, *table.delete(a,b)* removes the key (a, b) and the associated value from the table. Additionally, *mergeInCorpus(a,b)* names a method that merges all occurences of the syllable pair (a, b) in the text corpus.

4. Remove the respective entries from the hash table.
   $\forall\ (p_{i1}, p_{i2})\ \epsilon\ P$: table.delete$(t, (p_{i1}, p_{i2}))\}$

One can think of different stop criteria for this algorithm, e.g. perplexity or OOV rate based. We chose an OOV rate of 5% as the stop criterion. The JRTk can handle a maximum of 64k words in the recognition vocabulary. The OOV rate, however, is still below 1% for the resulting systems when the maximum vocabulary has been reached. Thus, in practice we use the vocabulary limit of 64k as stop criterion for the algorithm.

To preserve the eojeol boundaries we limit the unit merging process to unit pairs that lie in the same eojeol unit (*IntraEojeol*). But as the text data is not free of errors we consider a second variation where also units over eojeol boundaries may be merged (*InterEojeol*). This way, we also find out whether knowing the eojeol boundaries is an important source of information.

Two ways of selecting character pairs in step 2 of the above procedure are evaluated:

MergeAll Select *all* character pairs that produce the transition t:

$$P = \{(p_{i1}, p_{i2}) \mid \text{table.isKeyPair}(t, (p_{i1}, p_{i2}))\}$$

As the transition t between units disappears from the corpus, the acoustic confusability for this transition is eliminated.

MergeMax Select *the most frequent* character pair(s) that produce t:

$$\begin{aligned} P = \{(p_{i1}, p_{i2}) \mid\ & \text{table.isKeyPair}(t, (p_{i1}, p_{i2})) \wedge \\ & \text{table.value}(t, (p_{i1}, p_{i2})) \geq \\ & \text{table.value}(t, (p_{j1}, p_{j2})),\ \forall j \neq i\} \end{aligned}$$

With this approach, the transition t is not necessarily eliminated from the corpus. But, merging the most frequent character pair(s) reduces confusability and, in practice, many of the selected pairs are phrase-like pairs which to merge makes sense also from a language modeling point of view.

As it is not feasible to apply this algorithm to very large text corpora in RAM, the algorithm is split up into two passes: In the first pass we use the hash table to calculate a list of candidate pairs for merging. Then chunks of these candidates are merged in the large text corpus on disk. This process is repeated until a vocabulary of no more than 64k units is reached.

6.3 Speech Recognition Systems

Each of the *MergeAll* and the *MergeMax* approach are evaluated with both the *InterEojeol* and the *IntraEojeol* variation. This results in four systems. The naming convention for these systems is as follows: *MergeIntraAll* combines the *MergeAll* and *IntraEojeol* approaches. *MergeInterMax*, *MergeIntraAll* and *MergeIntraMax* are analogous.

The number of character pairs that were joined before the stop criterion was reached is 5,393 for *MergeIntraMax*, 19,951 for *MergeIntraAll*, 1,581 for *MergeInterMax* and 15,069 for *MergeInterAll*. See table 6.1. The *MergeMax* approach merges less unit pairs than the *MergeAll* approach. This is because the unit pairs merged with *MergeMax* have, on an average, a higher frequency in the text data than the *MergeAll* units and therefore the potential of creating new vocabulary units is higher for *MergeMax* pairs.

	Number of character pairs merged
MergeIntraMax	5,393
MergeIntraAll	19,951
MergeInterMax	1,581
MergeInterAll	15,069

Table 6.1: Number of character pairs that were merged.

Table 6.2 summarizes the corpus and vocabulary size of the four systems for the corpora *Train* and *PartChosun+Train*. The figures listed are in terms of units that result from the respective merging process. Interestingly, although *MergeMax* merges less different unit pairs than *MergeAll*, the *MergeMax* corpus size is smaller than that of the corresponding *MergeAll* system. Again, this is because the average frequency of each merged pair is higher in the *MergeMax* approach.

The OOV rates of the corpora *Test* and *Test-Utts* for the four systems are summarized in table 6.3. Both OOV rate and number of OOV words are calculated on both *Train* and *PartChosun+Train*. For all four systems the OOV rate using *PartChosun+Train* is around one percent. Such a low OOV rate is very good for large vocabulary speech recognition.

	PartChosun+ Train	Train
MergeIntraMax		
corpus size	4,032,575	174,387
vocabulary size	65,498	17,124
MergeIntraAll		
corpus size	4,707,499	203,623
vocabulary size	65,495	14,766
MergeInterMax		
corpus size	4,992,805	221,377
vocabulary size	65,479	10,924
MergeInterAll		
corpus size	5,466,419	239,158
vocabulary size	65,491	10,836

Table 6.2: Summary of merge based language model corpora.

The vocabulary growth on *PartChosun+Train* for all four merge based systems is shown in figure 6.1. The vocabulary grows significantly slower than on an eojeol based corpus. Also, the vocabulary grows faster for *MergeMax* than for *MergeAll*.

	Test	*Test-Utts*
MergeIntraMax		
Train	5.40% (1159)	6.08% (121)
PartChosun+Train	0.98% (210)	1.31% (26)
MergeIntraAll		
Train	4.61% (1147)	4.88% (111)
PartChosun+Train	1.14% (283)	1.14% (26)
MergeInterMax		
Train	2.77% (750)	2.78% (69)
PartChosun+Train	0.85% (231)	0.93% (23)
MergeInterAll		
Train	2.93% (850)	3.35% (89)
PartChosun+Train	0.96% (280)	1.13% (30)

Table 6.3: Summary of OOV rates and, in parantheses, OOV words.

The eight figures 6.2 to 6.5 and D.4 to D.7 show coverage characteristics of the four merge systems. They show for each system the self coverage of both *Train* and *PartChosun+Train* and their cross coverage on *Test* and *Test-Utts*. The coverage diagrams for the *Train* corpus are in the appendix. These diagrams show that the coverage for merge based units is more consistent than for eojeols; the self coverage and the cross coverage diagrams are

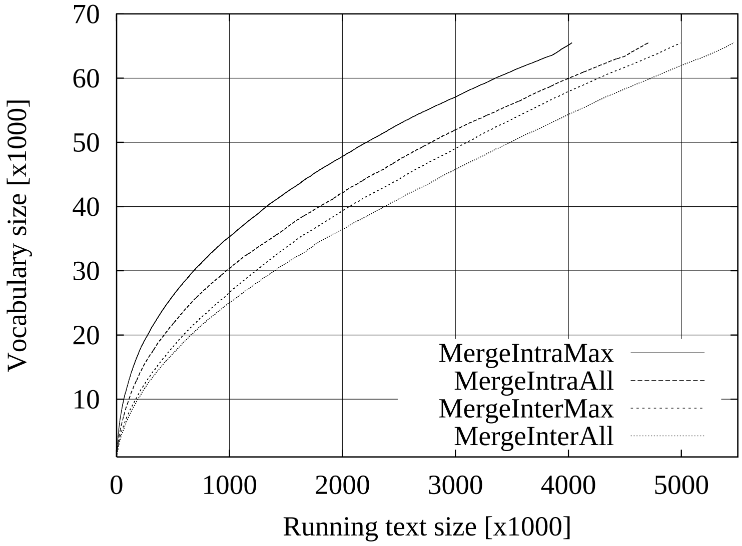

Figure 6.1: Vocabulary growth for the four merge based systems in corpus *PartChosun+Train*.

almost identical. For the eojeol units the cross coverage is up to five percent lower than the corresponding self coverage as can be seen from the diagrams 4.1 and 4.2. Comparing the figures shows us that coverage grows slightly slower in the *MergeAll* setup than in the *MergeMax* setup.

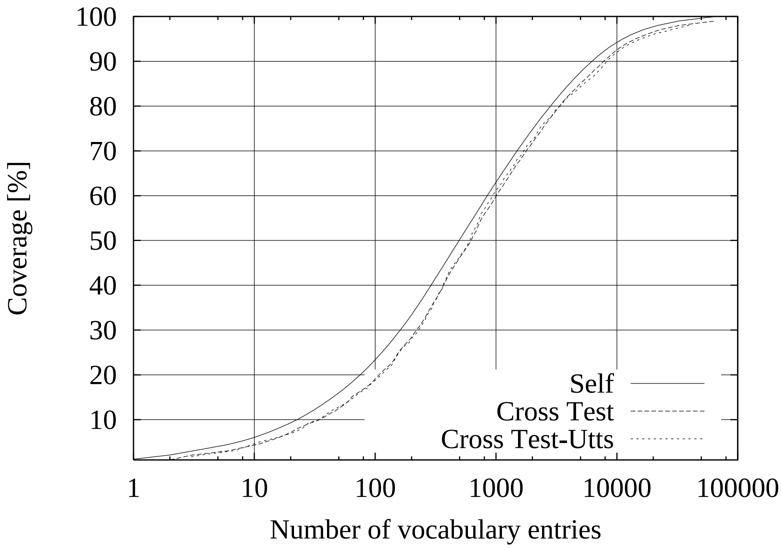

Figure 6.2: *MergeIntraMax*-based self coverage of *PartChosun+Train* and cross coverage of *Test* and *Test-Utts*.

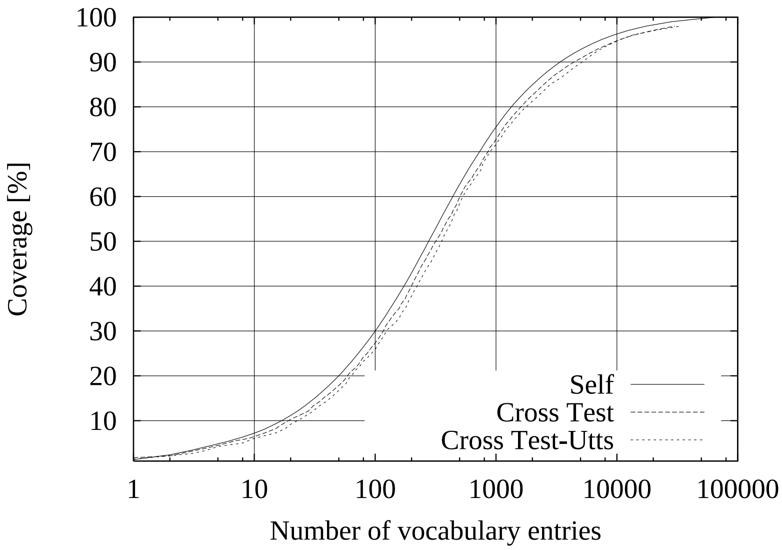

Figure 6.3: *MergeIntraAll*-based self coverage of *PartChosun+Train* and cross coverage of *Test* and *Test-Utts*.

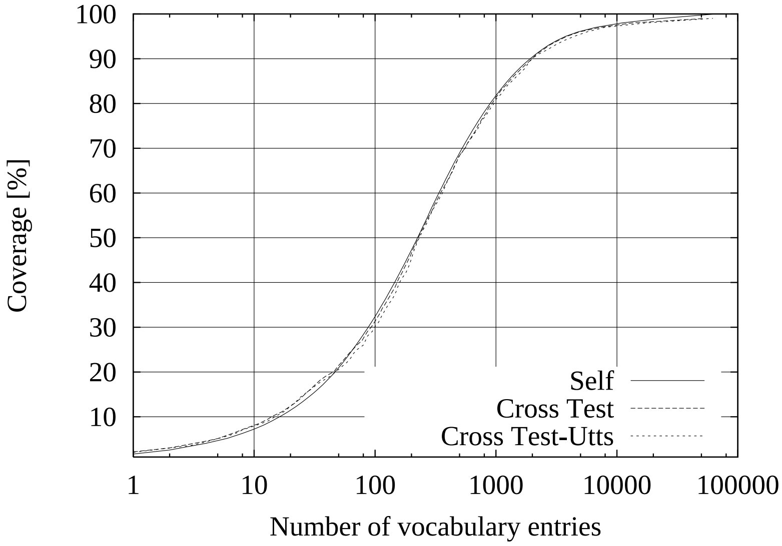

Figure 6.4: *MergeInterMax*-based self coverage of *PartChosun+Train* and cross coverage of *Test* and *Test-Utts*.

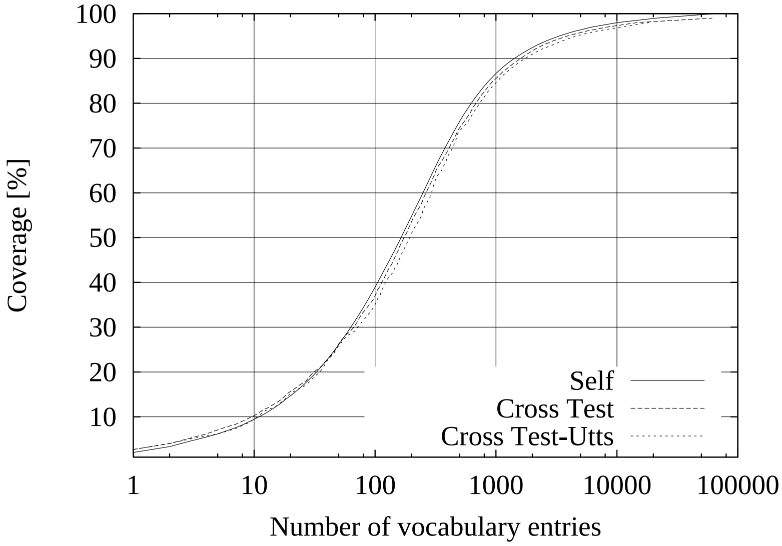

Figure 6.5: *MergeInterAll*-based self coverage of *PartChosun+Train* and cross coverage of *Test* and *Test-Utts*.

In addition to these four systems we also evaluated the *MergeIntraMax* approach, which performed the best of the four merge based systems, on the whole corpus *Chosun+Train*. This system will be referred to as *MergeIntraMaxWole*. With this approach 3,125 character pairs were merged. The characteristics of the resulting text corpora are as follows: *Chosun+Train* consists of 28,968,433 units and 65,328 vocabulary entries. The OOV rate and OOV words of *Test* are 0.12% and 28, respectively. For *Test-Utts* the according figures are 0.14% and 3. The *Train* corpus has a size of 196,185 units and a vocabulary size of 11,478. In this case the OOV rate and OOV words of *Test* are 2.84% and 682, respectively and for *Test-Utts* they are 3.01% and 67. Table 6.4 summarizes this information.

	Chosun+ Train	Train
Corpus characteristics		
corpus size	28,968,433	196,185
vocabulary size	65,328	11,478
OOV rates and OOV words		
Test	0.12% (28)	2.84% (682)
Test-Utts	0.14% (3)	3.01% (67)

Table 6.4: Summary of characteristics of *MergeIntraMax-Whole* corpora, and summary of OOV rates and, in parantheses, OOV words.

The vocabulary growth on *MergeIntraMaxWhole* is displayed in figure 6.6.

The figures 6.7 and D.8 show the coverage characteristics of the approach *MergeIntraMaxWhole* on the corpora *Chosun+Train* and *Train*.

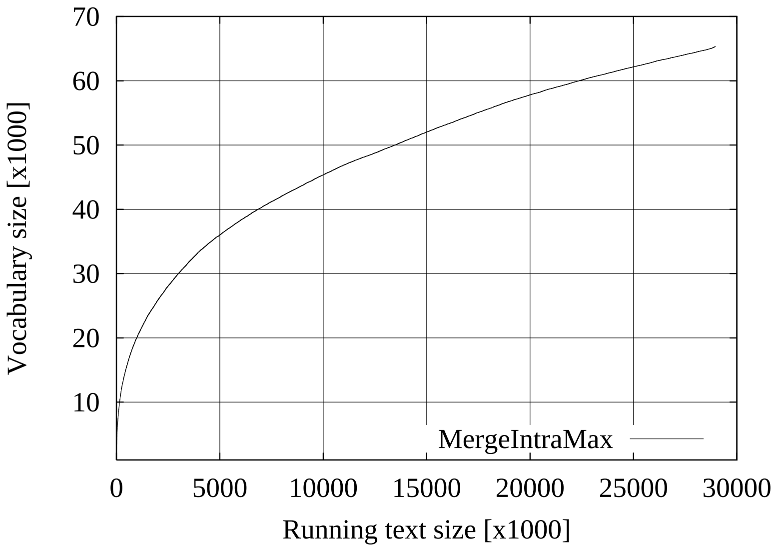

Figure 6.6: Vocabulary growth for *MergeIntraMaxWhole* based system in corpus *Chosun+Train*.

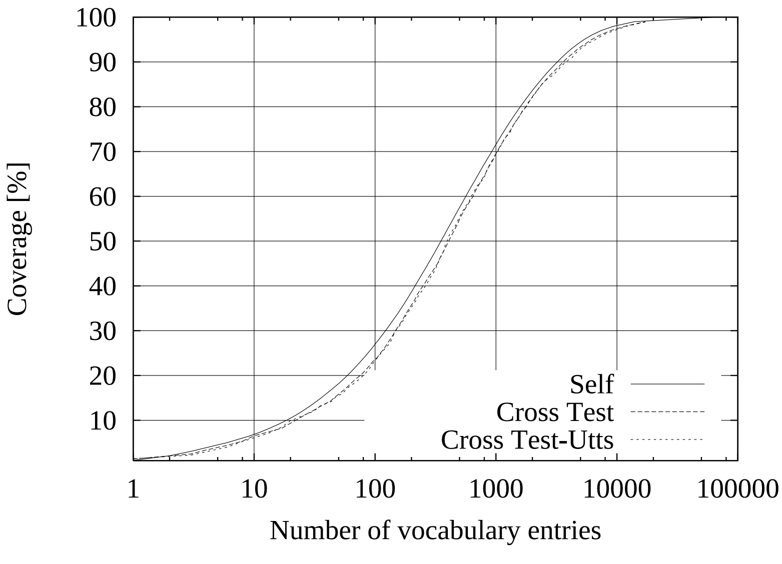

Figure 6.7: *MergeIntraMaxWhole*-based self coverage of *Chosun+Train* and cross coverage of *Test* and *Test-Utts*.

Chapter 7

Experiments

This chapter presents and discusses the results of the evaluated speech recognition systems. The necessary prerequisites for the experiments are discussed in section 7.1. Section 7.2 contains additional information on context dependent phone modeling. The experimental results are given in section 7.3.

7.1 Introduction

7.1.1 Recognition Accuracy

To determine the recognition performance of our systems, we measured the word error rate (WER), defined as:

$$\textit{Word Error Rate} = 100 \cdot \frac{\textit{substitutions + deletions + insertions}}{\textit{number of spoken words}}$$

As there is no straightforward Korean word unit error rates are given for several unit sets, namely eojeols, characters and phones. A value comparable to the word error rate in English is about the average of eojeol and character error rate. The following example illustrates the concept "error rate". The upper line in the example shows the transcription of an utterance, the lower line shows the recognized hypothesis.

동무는	언제	아버님에게		편지를	씁니까
동무는		아버님과	어머님에게	편지를	씁니까

동무는 is recognized correctly. 언제 causes a deletion error as it does not appear in the hypothesis. 아버님에게 is confused with 아버님과. 어머님에게 is an insertion error in the hypothesis. The remaining two eojeols are recognized correctly. With five reference eojeols in total this leaves us an eojeol based error rate of $100 \cdot \frac{1+1+1}{5}\% = 60\%$.

To determine the character based error rate we break up each eojeol into its character components and calculate the error rate of the resulting two sequences of characters. In our example that is:

동	무	는	언	제	아	버	님					에	게	편	지	를	씁	니	까
동	무	는			아	버	님	과	어	머	님	에	게	편	지	를	씁	니	까

A total of 14 characters is recognized correctly. Two characters are deleted, four are inserted. Thus, the character error rate is $100 \cdot \frac{0+2+4}{16}\% = 37.5\%$.

To calculate the phone based error rate of these two sentences we look up the phonetic transcription of each eojeol unit in the dictionary and align the two resulting phone strings:

D O NG M U N EU N EO N J E A B EO N I M E G E ...
D O NG M U N EU N A B EO N I M G O AE O M EO N I M E G E ...

...Ph iEO N J I R EU L SS EU M N I GG A

...Ph iEO N J I R EU L SS EU M N I GG A

The phone based error rate is $100 \cdot \frac{0+4+9}{36}\% = 36.11\%$.

7.1.2 Lattice Rescoring

Using Bayes' formula, acoustic probabilities and language model probabilities can be combined. Generally, the means and variances of these two classes of probabilities do not match and, as a consequence, either the acoustic or the language model side dominates the overall probability of a hypothesis. To compensate for this effect, the probabilities are weighted before they are multiplied. In addition, the JRTk decoder uses a word transition penalty q to control the length of the produced word sequence. The following variation of Bayes' formula is used:

$$P'(W \mid X) = \frac{P(X \mid W) \cdot P(W)^z \cdot q^{|W|}}{P(X)}$$

The exponent z is called the *language model weight*.

The use of z and q is very common in speech recognition research [44]. As it is not clear how to find the "optimal" (z, q) pair, the following strategy is used: A range of values is specified for both z and q. Then for each combination the overall recognition accuracy on the test set is calculated by rescoring the word lattice of each test utterance using these values. Finally, the pair (z_1, q_1) is picked which produces the highest recognition accuracy.

For our systems, we used the following heuristically determined ranges for z and q: $z \epsilon \{5, 10, \ldots, 55\}$ and $q \epsilon \{-30, -25, \ldots, 15\}$.

Our systems work with sub-eojeol units in the dictionary. However, we want to have a high-performance recognition on eojeol units. Thus, we must be able to determine which sub-eojeol units in a hypothesis have to be joined to form one eojeol unit. This issue is addressed as follows: each sub-eojeol unit that is inside an eojeol unit, as opposed to at the beginning, is prepended with a marker symbol, a dash. Then a sequence of sub-eojeol units can be transformed into a sequence of eojeols simply by connecting each "marked" unit to its predecessor. This applies to any type of sub-eojeol units we use; merge based units and morpheme units.

The lattice rescoring, i.e. determination of the "optimal" (z, q) pair, is then done as follows: for each combination of z and q we generate the eojeol versions of each sub-eojeol based reference and its respective hypothesis using the above strategy. On the resulting strings the eojeol word error rate can be calculated. We then pick the (z, q) pair that minimizes this eojeol error rate. For the speaker specific test results, minimal eojeol error rate is reported by determining a speaker-specific optimal (z, q) pair.

7.2 Context Modeling

Each phone is modeled context-dependently using a decision tree. These trees are generated using a set of 63 phonetically motivated context questions which are listed in table C.3 in the appendix.

In addition to the phonetically motivated questions we use two more questions that are related to the boundary characteristics of a phone. The first question asks whether a phone is at the boundary of a dictionary unit. This is a commonly used question in context-dependent modeling to capture cross-word effects. The second question asks whether a phone is at the left boundary of a dictionary unit which itself is not the left-most unit of an eojeol. This question gives the decoder component a mean to discriminate cross-unit models that are inside the same eojeol from cross-unit models over eojeol boundaries. In other words, this question helps the decoder to decide whether to stay within an eojeol unit or to start a new one.

7.3 Results And Discussion

7.3.1 Baseline

The complexity of a recognition task is generally measured in terms of its perplexity. The perplexity indicates the average word branching factor, i.e. the average number of words that can follow the current word. For an information theoretic discussion of the perplexity measure see [24]. Table 7.1 summarizes the perplexity information and also repeats the OOV rate for each system. These values are based on the corpora *PartChosun+Train* and *Test*. We see from the table that all merge based systems have significantly lower OOV rates than the morpheme based systems. The main reason is that the vocabulary size of the morpheme based corpora is much higher than 64k, see table 5.1. But only the most frequent 64k words can be used in the recognizer. On the other side, the merge based corpora contain no more than 64k different units. Unfortunately, the perplexity values displayed in table 7.1 are not directly comparable as they are calculated on different unit sets.

Table 7.2 shows the error rates for the four merge based systems and the two

	OOV	Perplexity
MergeIntraMax	0.943%	481.7
MergeIntraAll	1.102%	206.3
MergeInterMax	0.830%	149.5
MergeInterAll	0.939%	104.7
Morph	2.371%	209.2
MorphTag	2.784%	216.4

Table 7.1: Recognition task complexities on *Chosun+Train*.

morpheme based systems. Lattice rescoring is done on the eojeol level. Character and phone error rate is then determined as described in section 7.1.1 with the same (z, q) pair. Note, that the reported character and phone error rates do not necessarily have to be maximal using this evaluation scheme. In the first column, the table shows the error rates on the whole test set. The columns labelled "1" to "10" show the speaker-specific error rates. These results were obtained using the corpus data *PartChosun+Train*.

We see from table 7.2 that the plain morpheme system *Morph* is not outperformed by *MorphTag*. The appended POS-tag gives the *MorphTag* units additional information, but this also increases the size of the vocabulary which leads, due to the 64k test vocabulary limitation, to a higher OOV rate than for the system *Morph*. Also, the language model is less reliable as more 3-gram models have to be estimated from the same amount of text data. In combination, these effects make both morpheme systems perform equally well.

The performance of *MergeInterMax* and *MergeInterAll* is significantly worse than that of either the *IntraEojeol* based systems or the morpheme systems. This tells us that the eojeol boundaries are an important structural element and that merging units over eojeol boundaries seems to destroy this structure, leaving us with units which are not well suited for 3-gram modeling.

We see from table 7.2 that *MergeMax* outperforms *MergeAll* in both the *InterEojeol* and the *IntraEojeol* case. There are two reasons for this behavior: First, the character pair which is merged is in many cases part of a suffix combination or a multiple-character stem that occurs very frequently. These

	All	1	2	3	4	5	6	7	8	9	10
MergeIntraMax											
Phone ER	16.2	19.3	21.3	12.3	10.3	13.4	16.0	12.6	18.1	15.5	27.4
Character ER	24.2	27.4	28.6	17.0	17.0	19.4	23.2	20.1	26.8	22.8	37.1
Eojeol ER	39.8	41.4	43.2	24.4	31.9	35.0	37.3	32.5	38.6	38.8	64.8
MergeIntraAll											
Phone ER	17.9	20.9	21.3	12.5	11.1	14.4	15.5	13.9	17.8	17.0	27.2
Character ER	25.9	30.4	28.4	18.8	17.9	21.4	22.8	21.6	26.8	24.4	38.6
Eojeol ER	44.2	48.1	48.6	31.5	34.8	37.5	39.1	36.8	38.6	42.9	59.2
MergeInterMax											
Phone ER	19.9	21.7	22.8	14.7	13.0	14.3	20.1	18.1	20.5	17.4	23.2
Character ER	29.5	32.7	35.3	21.7	20.9	19.9	30.5	28.4	32.1	25.2	32.2
Eojeol ER	48.6	52.6	60.6	33.1	37.7	41.7	46.4	45.6	49.1	42.1	57.7
MergeInterAll											
Phone ER	20.2	21.3	22.9	12.2	16.6	11.4	16.1	20.4	27.6	19.7	24.6
Character ER	29.7	31.0	32.4	17.3	25.4	17.4	24.4	33.3	38.0	29.4	37.6
Eojeol ER	50.0	50.0	60.2	32.3	42.8	33.3	40.9	52.6	57.9	50.0	57.7
Morph											
Phone ER	15.6	20.0	17.4	10.8	11.6	10.6	13.9	13.1	20.2	12.9	20.8
Character ER	22.4	29.5	24.9	15.5	17.9	14.5	21.5	18.2	26.8	18.9	30.7
Eojeol ER	38.1	42.9	43.1	26.0	30.4	28.3	35.5	32.5	44.7	31.7	52.1
MorphTag											
Phone ER	15.8	17.3	19.0	13.9	12.6	10.3	14.0	14.5	17.5	14.4	24.8
Character ER	23.3	27.1	27.1	19.6	20.9	15.1	21.2	22.5	24.0	20.7	34.2
Eojeol ER	38.2	39.1	41.4	30.7	34.1	29.2	33.6	38.6	41.2	33.3	59.2

Table 7.2: Summary of recognition error rates, %.

are units that would be combined in a phrase based language model and merging them thus makes sense from a language modeling perspective. The *MergeMax* approach only merges such max-pairs, therefore generating more phrase-like units. This seems to increase the language model suitability of the resulting units. Second, merging every pair that produces a pronunciation transition, as done in the *MergeAll* approach, eliminates this transition totally. But at the same time this increases the size of the vocabulary, decreasing the number of further transitions to merge.

The most interesting result that we can extract from table 7.2 is that the system *MergeIntraMax* and the morpheme based system perform about equally well. The speaker-specific error rates show that on some speakers the morpheme systems perform better while on others they are worse than *MergeIntraMax*. In fact, the total eojeol error rate of the morpheme system is about 1.5% smaller than for the merge based system. But as the eojeol error rate is a very sensitive measure[1], we can not imply that the morpheme systems

[1] Consider for example the eojeol sequence “a-b-c d-e f-g”, where the letters symbolize

perform better than *MergeIntraMax* in general.

7.3.2 Pronunciation Variant LM

We can further improve the recognition performance by using a "pronunciation-variant based language model". As said in section 4.3, the pronunciation of a unit depends on its context. Different pronunciations of a unit are handled as pronunciation variants in the dictionary, distinguished by appended variant indizes. To estimate a variant based language model, each unit in the text corpus gets appended the index of its pronunciation variant in the dictionary. The 3-gram model is calculated on the resulting text corpus. Thus, pronunciation variants of a unit are considered different language model units. One disadvantage is that the number of units is increased and therefore more 3-grams have to be estimated from the same amount of text material. In addition, these variant language models have slightly higher perplexities and OOV rates than their counterparts that do not distinguish pronunciation variants. Each pronunciation variant of a unit is considered a distinctive vocabulary unit in the recognizer. With the 64k vocabulary size limitation less different words can be used in the vocabulary and as a consequence the OOV rate will increase. We evaluated this idea on the systems *MergeIntraMax* and *MorphTag*. The task perplexity and the OOV rate of these systems are given in table 7.3.

	OOV	Perplexity
MergeIntraMax	1.518%	498.3
MorphTag	3.575%	224.4

Table 7.3: Recognition task complexities on *Chosun+Train*, pronunciation variants not mapped on baseform in language model.

Table 7.4 shows the performance results. Introducing a variant-LM decreases the eojeol error rate of the system *MergeIntraMax* to 35.6%, an error reduc-

syllable components. While both the character and phone error rate for "a-b c-d-e f-g" are 0%, the eojeol error rate would be 66%. Small errors can yield big differences in the eojeol error rate.

tion of 10.5%. For *MorphTag* the eojeol error rate is reduced by 15.4% to 32.3%.

	All	1	2	3	4	5	6	7	8	9	10
MergeIntraMax											
Phone ER	14.3	15.2	16.4	8.9	10.7	10.1	13.4	9.6	18.3	13.0	22.7
Character ER	21.2	24.2	24.3	11.4	17.0	15.7	20.3	15.1	24.9	17.6	33.2
Eojeol ER	35.6	36.1	41.5	21.3	30.4	25.8	34.5	27.2	36.0	29.5	59.2
MorphTag											
Phone ER	13.5	14.2	18.5	10.0	9.7	7.5	12.0	8.2	12.0	11.1	22.2
Character ER	19.8	21.2	26.9	14.7	14.5	10.8	16.4	12.7	16.8	16.0	33.7
Eojeol ER	32.3	33.1	40.6	25.2	24.6	21.7	26.4	22.8	28.1	30.9	57.7

Table 7.4: Summary of recognition error rates using a variant-LM, %.

7.3.3 More Data

In addition, we evaluated the two above systems, using the variant-LM, on the full text corpus *Chosun+Train*. The resulting systems are called *MergeIntraMaxWhole* and *MorphTagWhole*. Using a larger amount of text data improves the robustness of the language model. To keep the vocabulary size of the merge system at 64k, the merging has to be done on the bigger text corpus again. The recognition task characteristics of the resulting systems are shown in table 7.5. The OOV rate and perplexity values show us that the merge based system profits a lot from the enlarged amount of text data. The OOV rate is reduced by more than one percent and the perplexity is almost reduced by a factor of five. But again, the perplexity values cannot be compared directly as the unit sets of the two *MergeIntraMax* systems differ. This is because less units can be merged on the big corpus before a 64k vocabulary is reached. The OOV rate of *MorphTag* remains around 3.5% but the perplexity increases by a factor of two. The reason for this is the increase in vocabulary size by a factor of about four.

The performance results of *MergeIntraMaxWhole* and *MorphTagWhole* are listed in table 7.6. The increased amount of text data results in an eojeol error rate of 28.0% for *MergeIntraMaxWhole*, an error reduction of 21.3%. For *MorphTagWhole* the eojeol error rate is reduced by 3.1% to 31.3%. While

	OOV	Perplexity
MergeIntraMaxWhole	0.233%	136.6
MorphWhole	2.886%	143.1
MorphTagWhole	3.513%	485.9

Table 7.5: Recognition task characteristics.

the merge based system profits from the increased amount of text data, the improvement for the morpheme system is not very significant. The main reason for this difference in improvement should be that the vocabulary for *MergeIntraMaxWhole* is kept at 64k resulting in a very low OOV rate of 0.23%.

	All	1	2	3	4	5	6	7	8	9	10
MergeIntraMaxWhole											
Phone ER	11.6	14.7	14.7	7.3	9.1	9.2	12.4	10.3	12.3	9.1	19.9
Character ER	16.4	20.4	21.0	9.7	14.8	12.5	17.4	15.1	17.1	12.3	29.7
Eojeol ER	28.0	29.3	33.7	16.4	26.1	22.5	27.3	25.4	24.6	22.3	47.9
MorphTagWhole											
Phone ER	12.2	10.6	16.8	9.4	11.2	6.4	12.9	10.8	13.5	9.1	18.5
Character ER	18.1	14.7	24.5	13.5	18.2	10.0	18.6	18.5	18.4	12.9	26.2
Eojeol ER	31.3	23.1	38.9	25.0	32.1	19.8	31.8	29.8	29.8	24.5	47.9

Table 7.6: Summary of recognition error rates on *Chosun+Train*, %.

7.3.4 Corrected Speech Database

An analysis of the recognition errors revealed that one major type of error was made, namely the deletion of one syllable in the beginning or the end of an utterance. As a consequence, the segmentation of the audio waveform files was checked and it was found that most of them were segmented too sharply. In many cases, as much as a whole syllable was cut away. Subsequently, the whole database was resegmented. New labels were created with the repaired database and the acoustic models were retrained on these new labels. With the new acoustic models we repeated the last experiment on the new database. In addition to *MergeIntraMaxWhole* and *MorphTagWhole* the system *MorphWhole* was also evaluated.

Table 7.7 shows the performance results. Working on the corrected database decreases the eojeol error rate of *MergeIntraMaxWhole* to 24.6%, an error reduction of 12.1%. For *MorphTag* the eojeol error rate is reduced by 4.1% to 30.0%. *MorphWhole* has an eojeol error rate of 24.0%

	All	1	2	3	4	5	6	7	8	9	10
MergeIntraMaxWhole											
Phone ER	9.9	11.4	11.3	8.0	6.7	7.5	11.8	8.2	9.6	6.6	12.9
Character ER	14.5	18.6	15.5	11.4	10.1	12.0	16.4	11.1	12.8	10.2	20.3
Eojeol ER	24.6	24.8	29.1	16.5	20.3	21.7	26.4	21.1	18.4	18.6	29.6
MorphWhole											
Phone ER	9.4	5.8	13.0	4.8	6.5	9.0	11.7	8.9	8.9	7.0	15.9
Character ER	13.0	7.4	18.2	6.7	10.3	11.4	18.6	11.1	10.6	10.0	22.3
Eojeol ER	24.0	13.4	34.3	14.1	16.7	20.8	28.2	18.4	17.5	18.0	38.0
MorphTagWhole											
Phone ER	10.7	6.7	14.2	8.2	9.7	8.5	11.5	7.5	9.4	8.2	17.1
Character ER	16.2	11.2	21.8	12.3	14.0	13.1	18.0	13.0	12.5	12.1	21.8
Eojeol ER	30.0	19.4	37.1	21.9	27.0	27.5	35.5	27.2	21.9	21.1	42.3

Table 7.7: Summary of recognition error rates on *Chosun+Train* with corrected speech database, %.

7.3.5 Discussion

The best merge based baseline system, *MergeIntraMax*, has an eojeol error rate of 39.8%. The error rate of the system *MorphTag* is 1.6% lower, 38.2%. Introducing pronunciation-variant based language models reduces the eojeol error rate of *MergeIntraMax* by 10.5% to 35.6%. The error rate of *MorphTag* using a variant based language model is 32.3%, an error reduction of 15.4%. Increasing the amount of training data for the language model yields a further error reduction of 21.3% for *MergeIntraMax*, but only 3.1% for *MorphTag*. The resulting eojeol eror rates are 28.0% for *MergeIntraMax* and 31.3% for *MorphTag*. We found out that most utterances in the speech database were segmented too sharply. As a consequence, we corrected the segment boundaries of each utterance and used the repaired database to retrain the acoustic models. Using the variant-LM and the large text corpus on the new database, *MergeIntraMax* has an eojeol error rate of 24.6%, *MorphTag* has an error rate of 30.0%. This is an error reduction of 12.1% and

4.1%, respectively. To summarize, we reduced the eojeol error rate of 39.8% of the baseline *MergeIntraMax* system to 24.6%, a total error reduction of 38.2%. The eojeol error rate of the baseline *MorphTag* system, which was 38.2%, was reduced by 21.5% to an error rate of 30.0%. This information is summarized in table 7.8

The system *Morph* has an eojeol error rate of 24.0% on the corrected database. It outperforms the system *MorphTag* due to two reasons. First, the vocabulary size of *MorphTag* is significantly higher than the one of *Morph*. As a consequence, *MorphTag* has a higher OOV rate when its vocabulary is limited to 64k. Also, the *MorphTag* language model is less robust than the *Morph* language model as more 3-grams have to be estimated from the same amount of training data.

	MergeIntraMax		*MorphTag*	
	Eojeol Error Rate	Error Reduction	Eojeol Error Rate	Error Reduction
Baseline	39.8	-	38.2	-
Variant-LM	35.6	10.5	32.3	15.4
More Data	28.0	21.3	31.3	3.1
Corrected Database	24.6	12.1	30.0	4.1

Table 7.8: Summary of the performance of the recognition systems.

All these experiments consistently show that the dictionary units determined by the data-driven approach *MergeIntraMax* are no worse than the morpheme units determined by a complex morpheme analyzing system.

Chapter 8

Conclusions

8.1 Conclusions

The "agglutinating" nature of the Korean language makes the choice of appropriate dictionary units for a large vocabulary speech recognition system difficult. Eojeols, the units that result from the agglutination process, are highly inappropriate. They have an inherent severe OOV problem and show a linear with task size vocabulary growth rate. Sub-eojeol units must therefore be determined. A natural choice of sub-eojeol unit is the morpheme as each eojeol is composed of a sequence of morphemic components. The development of a system that automatically splits eojeol units into their morpheme components is very expensive, involving a lot of *apriori* expert knowledge on the morphological structure of the Korean language.

This work describes a new approach to finding appropriate dictionary units. Because morpheme units are not proven to be the optimal choice of sub-eojeol units from a language modeling perspective we discarded the morphemic structure and approached the problem from an acoustic, data-driven perspective instead. We developed an approach which, in a first pass, splits each eojeol into its syllable components. Then, iteratively, syllable pairs are merged in order to reduce acoustic confusability of syllable transitions.

Our results show that this data-driven approach generates units that are equally suitable for a large vocabulary speech recognition dictionary as the morpheme units. As a consequence, we do not have to rely on complex expert systems for the unit determination process. It can be done automatically using a data-driven approach.

8.2 Future Work

To further improve the performance of these speech recognition systems, a number of possibilities might be explored:

- *"LM-related" merging criterion:*
  A language model related extension of the function that chooses the unit pairs to merge could improve the quality of the determined unit set in terms of language model suitability. A simple example of such an extension would be the integration of *mutual information*.
  It would also be interesting to compare the unit merging approach to a purely LM motivated approach where iteratively the most frequent pair of syllables is merged, without taking into account the frequency of phone transitions.

- *"More Data":*
  A famous expression in the area of pattern recognition is: "There is no data like more data". One way to improve the recognition performance would be to collect more speech data for the acoustic models. Not only can the models be estimated more reliably but the system also increases its speaker independent robustness by learning more about "new speakers". Analogous, more text data would improve the quality of the statistical language model.

- *Higher-order LMs:*
  The employment of higher order n-gram language models ($n > 3$) would very likely yield an increase in performance. As the language model is based on sub-eojeol units, it is not always guaranteed that the 3-gram provides enough range to ensure valid eojeol unit connection.

- *System Tuning:*
  A general way to improve the performance of a speech recognition system is to tune its parameters. This would include the number of polyphone models, the dimension of the feature vector, the number of Gaussians and also variations in the preprocessing.

Appendix A

Transcription Systems for 한글

한글	McCune-Reischauer	Yale	North Korea	South Korea	hcode
ㅂ	p,b	p	p	b	p
ㅍ	p'	ph	ph	p	ph
ㅃ	pp	pp	pp	bb	pp
ㄷ	t,d	t	t	d	t
ㅌ	t'	th	th	t	th
ㄸ	tt	tt	tt	dd	tt
ㅅ	s	s	s	s	s
ㅆ	ss	ss	ss	ss	ss
ㅈ	ch,j	c	ts	j	c
ㅊ	ch'	ch	tsh	ch	ch
ㅉ	tch	cc	tss	jj	cc
ㄱ	k,g	k	k	g	k
ㅋ	k'	kh	kh	k	kh
ㄲ	kk	kk	kk	gg	kk
ㅁ	m	m	m	m	m
ㄴ	n	n	n	n	n
ㅇ	ng	ng	ng	ng	ng
ㅎ	h	h	h	h	h
ㄹ	l,r	l	r	l,r	l,r
ㅣ	i	i	i	i	i

Table A.1: Transcription systems for the 한글 consonants.

한글	McCune-Reischauer	Yale	North Korea	South Korea	hcode
ㅣ	i	i	i	i	i
ㅟ	wi	wi	wi	wi	wi
ㅔ	e	ey	e	e	e
ㅖ	ye	yey	ye	ye	ye
ㅞ	we	wey	we	we	we
ㅚ	oe	oy	oi	oe	oe
ㅐ	ae	ay	ai	ae	ae
ㅒ	yae	yay	yai	yae	yae
ㅙ	wae	way	wai	wae	wae
ㅡ	ŭ	u	ŭ	eu	eu
ㅓ	ŏ	e	ŏ	eo	eo
ㅕ	yŏ	ye	yŏ	yeo	yeo
ㅝ	wŏ	we	wŏ	weo	weo
ㅏ	a	a	a	a	a
ㅑ	ya	ya	ya	ya	ya
ㅘ	wa	wa	wa	wa	wa
ㅜ	u	wu	u	u	u
ㅠ	yu	yu	yu	yu	yu
ㅗ	o	o	o	o	o
ㅛ	yo	yo	yo	yo	yo
ㅢ	ui	uy	ui	eui	yi

Table A.2: Transcription systems for the 한글 vowels.

Appendix B

Text Corpus Mappings

Acronym	한글	Acronym	한글
3D	쓰리디	IBM	아이비엠
ABC	에이비씨	ISDN	아이에스디엔
ADAC	에이디에이씨	JPEG	제이펙
AI	에이아이	LAPD	엘에이피디
AMD	에이엠디	MBA	엠비에이
ATM	에이티엠	MRI	엠알아이
AT&T	에이티앤드티	NYSE	엔와이에스이
AWACS	어웩스	OEM	오이엠
BASF	비에이에스에프	PCMCIA	피씨엠씨아이에이
CDU/CSU	씨디유 씨에스유	PDA	피디에이
CeBIT	씨비트	R&D	알앤드디
CEO	씨이오	RAM	램
CGI	씨지아이	TOEFL	토플
CIA	씨아이에이	UCLA	유씨엘에이
CMOS	씨모스	UEFA	유이파
CPU	씨피유	UN	유엔
DAEWOO	대우	UNESCO	유네스코
EPROM	이프롬	UNICEF	유니쎄프
FBI	에프비아이	US	유에스
FIFA	피파	WTO	더블유티오
HBO	에이치비오	YMCA	와이엠씨에이

Table B.1: Some examples of acronym mappings.

Descriptor	한글	Descriptor	한글
%	퍼센트	m^3	세제곱미터
cc	씨씨	m^2	제킬로미터
cm	센티미터	m	미터
dB	데시벨	MB	메가바이트
g	그람	MHz	메가헤르쯔
GB	기가바이트	mm	밀리미터
ha	헥타	mW	밀리와트
Hz	헤르쯔	kW	킬로와트
kbps	킬로비피에스	pH	피에이치
kg	킬로그람	ppm	피피엠
kHz	킬로헤르쯔	t	톤
km	킬로미터	V	볼트
kV	킬로볼트		

Table B.2: Mapping table for units.

Appendix C

Janus Toolkit

C.1 Phoneme Models

IPA-symbol	Janus-Phone	Example
a	A	sh**ah**
ɛ	AE	c**a**t
e	E	b**e**t
i	I	**i**nn
o	O	**o**il
ə	EO	**a**bout
ø	OE	German G**oe**the, but also **we**t
u	U	g**oo**se, but shorter (rounded lips)
ɨ	EU	brok**e**n (unrounded lips)
y	UE	German f**ü**nf
ɨi	euI	sq**uee**ze pedantically as [ɨi] but usually initial as [ɨ], in other position as [i], and as particle meaning "of" as [e]
ia	iA	**ya**rn
ie	iE	**ye**llow
iə	iEO	**ye**rba
io	iO	**yo**ke
iu	iU	**you**th
oa	oA	**wa**sh
uə	uEO	**wo**nderful

Table C.1: Korean vowel models.

IPA-Symbol	Janus-Phone	Example and explanation
č	CHh	hit**chh**ike, strong aspiration
ǰ	J	re**j**oice, fully voiced
č'	JJ	pi**tch**er, tight throated release
s	S	**s**ing, weaker than its English counterpart
s'	SS	a**ss**ail, with tension in the throat and tongue
m	M	**m**omentum
n	N	**n**ewtonian, tongue tip behind upper teeth
ŋ	NG	si**ng**ing
h	H	**h**at
p	Ph	u**ph**ill, strong aspiration
b	B	o**b**ey, fully voiced
p˺	p	sto**p**, unreleased
p'	BB	s**p**in, tight throated release
t	Th	ho**th**ouse, strong aspiration
d	D	a**d**o, fully voiced
t˺	t	ye**t**, unreleased
t'	DD	s**t**ay, tight throated release
k	Kh	blo**ckh**ead, strong aspiration
g	G	a**g**o, fully voiced
k˺	k	ti**c**, unreleased
k'	GG	s**k**y, tight throated release
ɾ	R	ve**r**y (in British English)
l	L	tai**l**

Table C.2: Korean consonant models.

C.2 Phoneme Context Question Sets

CONSONANTS	Ph B BB p Th D DD t Kh G GG k CHh J JJ M N NG R L S SS H		
VOWEL	A EO O U I EU AE E OE UE iA iEO iO iU iE oA uEO euI		
VCD	B BB D DD G GG J JJ NG M N R L		
VELAR	Kh G GG k NG	SILENCE	SIL
BILABIAL	Ph B BB p	NOISE	+hGH
PALATAL	CHh J JJ	NASAL	M N NG
AFFRICATE	CHh J JJ	LATERALAPPR	L
EMPHASED	BB DD SS JJ GG	FRICATIVE	S SS H
ASPIRED	Ph Th Kh CHh	FRIC-ALVEO	S
PLOS-VCD	B BB D DD G GG	FRIC-GLOTTAL	H
PLOS-UNVCD	Ph Th Kh	EMPH-UNVCD	SS
CLOSE-VOW	I UE EU U	EMPH-BILABIAL	BB
EMPH-VCD	BB DD GG JJ	EMPH-VELAR	GG
VOW-I	I iA iEO iO iU iE	EMPH-ALVEO	DD SS
ALVEO-VCD	D DD N R L	EMPH-PALATAL	JJ
ALVEO-UNVCD	Th t S SS	IMPL-BILABIAL	p
DIPH-I	iA iEO iO iU iE	IMPL-ALVEO	t
ROUND	O U OE UE	IMPL-VELAR	k
UNROUND	EO EU I A AE E	BILABIAL-VCD	B BB M
FRONT-VOW	I E AE A UE OE	BILABIAL-UNVCD	Ph p
BACK-VOW	EU EO O U	PALATAL-VCD	J JJ
VOW-U	U UE uEO	PALATAL-UNVCD	CHh
ALVEO	Th D DD t N R L S SS	VELAR-UNVCD	Kh k
PLOSIVE	Ph B BB Th D DD Kh G GG	DIPH-O	oA
TRILL	R	DIPH-U	uEO
UNVCD	Ph p Th t Kh k CHh S SS H	DIPH-EU	euI
DIPHTHONG	iA iEO iO iU iE oA uEO euI	DIPH-DOWN	euI
DIPH-UP	iA iEO iO iU iE oA uEO	OPENMID-VOW	AE EO
VOW-O	O OE oA	OPEN-VOW	A
PLOS-BILABIAL	Ph B BB	VOW-EU	EU euI
PLOS-ALVEO	Th D DD	AFFR-VCD	J JJ
PLOS-VELAR	Kh G GG	AFFR-UNVCD	CHh
VELAR-VCD	G GG NG	IMPLOSIVE	p t k
GLOTTAL	H	CLOSEMID-VOW	E OE O

Table C.3: Phone sets used for decision tree based context dependent phone modeling.

Appendix D

Coverage

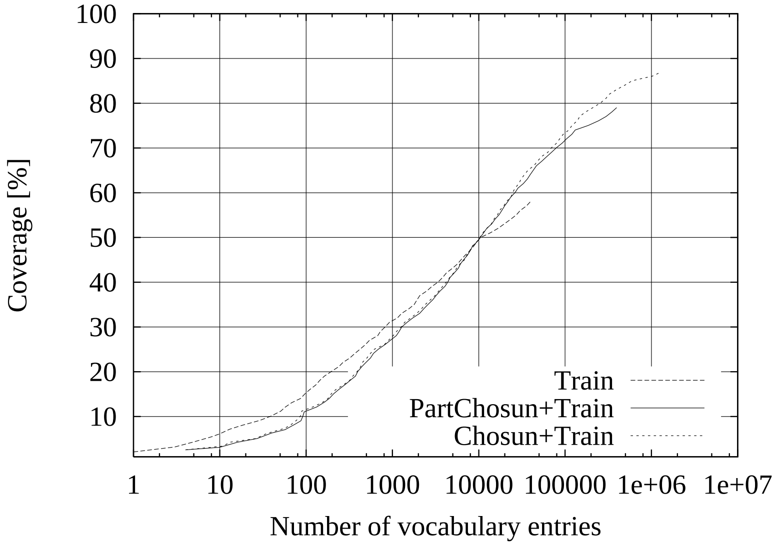

Figure D.1: Eojeol-based cross coverage of *Test-Utts* with different language model corpora.

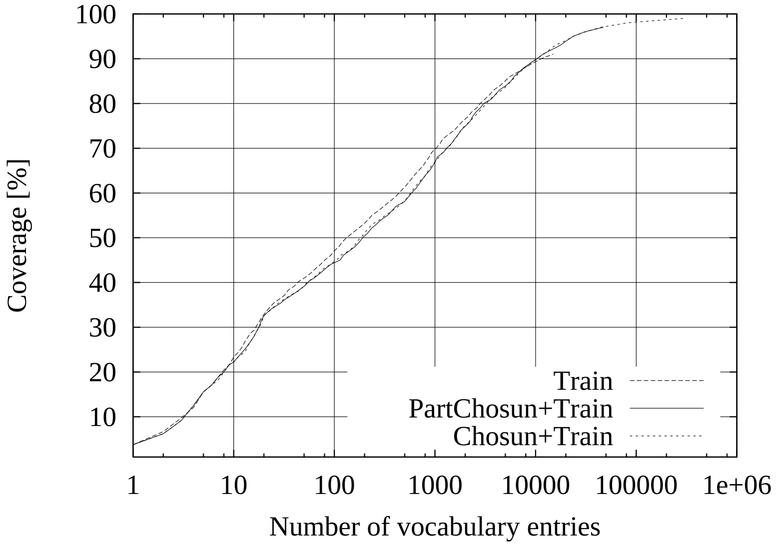

Figure D.2: *Morph*-based cross coverage of *Test-Utts* with different language model corpora.

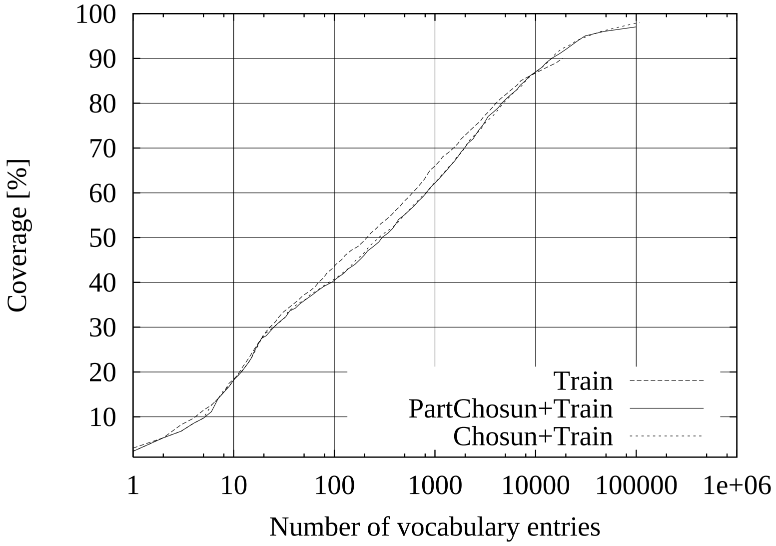

Figure D.3: *MorphTag*-based cross coverage of *Test-Utts* with different language model corpora.

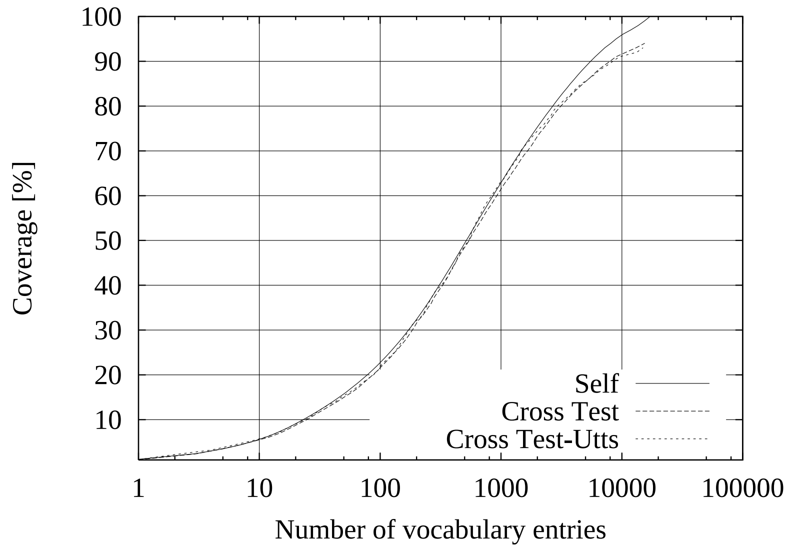

Figure D.4: *MergeIntraMax*-based self coverage of *Train* and cross coverage of *Test* and *Test-Utts*.

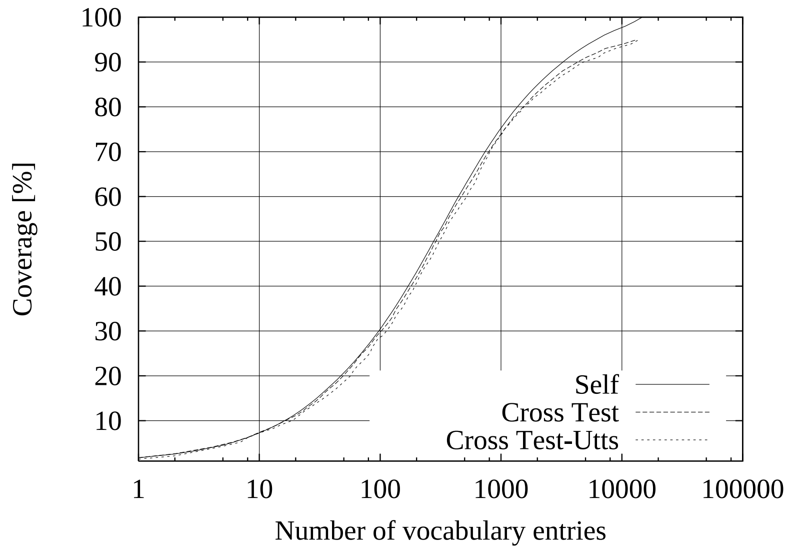

Figure D.5: *MergeIntraAll*-based self coverage of *Train* and cross coverage of *Test* and *Test-Utts*.

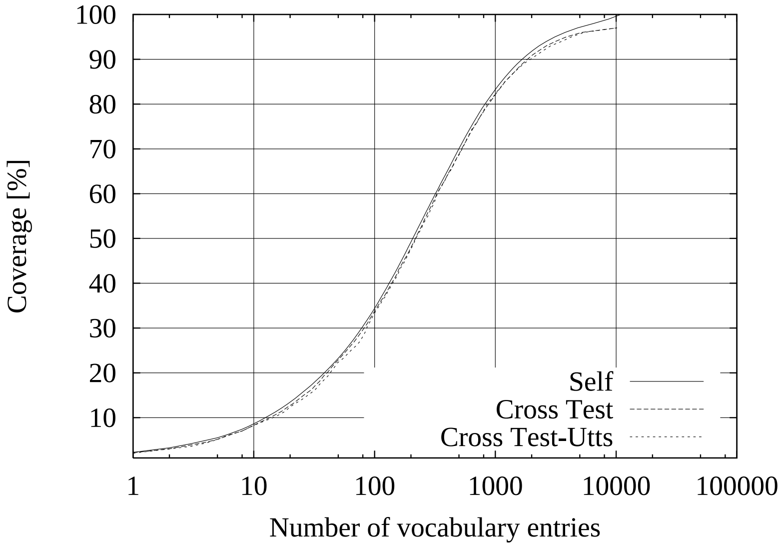

Figure D.6: *MergeInterMax*-based self coverage of *Train* and cross coverage of *Test* and *Test-Utts*.

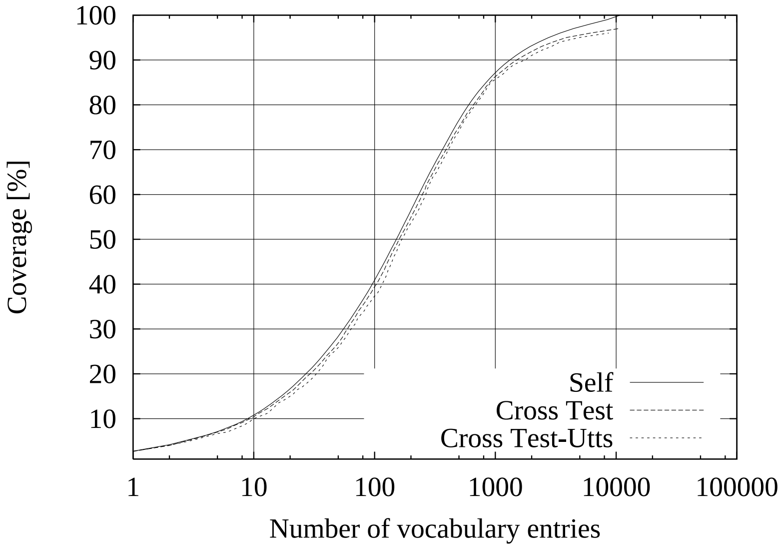

Figure D.7: *MergeInterAll*-based self coverage of *Train* and cross coverage of *Test* and *Test-Utts*.

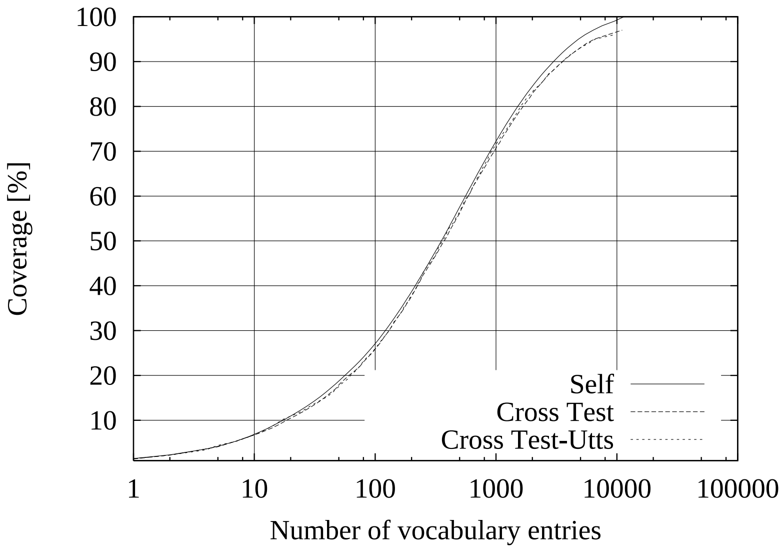

Figure D.8: *MergeIntraMaxWhole*-based self coverage of *Train* and cross coverage of *Test* and *Test-Utts*.

Bibliography

[1] International Phonetic Association. *Handbook of the International Phonetic Association.* Cambridge University Press, 1999.

[2] X. Aubert, R. Haeb-Umbach, and H. Ney. Continuous Mixture Densities and Linear Discriminant Analysis for Improved Context-Dependent Acoustic Models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 648–651. IEEE Signal Processing Society, IEEE; New York, NY, April 1993.

[3] L.R. Bahl, R. Bakis, P.S. Cohen, A.G. Cole, F. Jelinek, B.L. Lewis, and R.L. Mercer. Further Results on the Recognition of a Continuously Read Natural Corpus. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 872–876. IEEE, IEEE; New York, NY, April 1980.

[4] L.E. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes. *Inequalities*, (3):1–8, 1972.

[5] G. Boulianne and P. Kenny. Optimal Tying of HMM Mixture Densities Using Decision Trees. In *Proceedings of the International Conference on Spoken Language Processing, ICSLP 96*, volume 1, pages 350–353. IEEE Signal Processing Society, IEEE; New York, NY, October 1996.

[6] E.O. Brigham. *FFT : schnelle Fourier-Transformation.* R. Oldenburg Verlag, 1995.

[7] J.C. Buckow. Klassifizierung und Erkennung von Sprachsegmenten. Master's thesis, Universität Karlsruhe, October 1996.

[8] P.A. Chou. Optimal Partitioning for Classifications and Regression Trees. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 13(4):340–354, April 1991.

[9] K.Y.O. Dowling. *Korean Phrasebook.* Lonely Planet, 1995.

[10] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, 1973.

[11] E. Eide and H. Gish. A Parametric Approach to Vocal Tract Length Normalization. In *Proceedings of the IEEE International Conference On Acoustics, Speech And Signal Processing*, volume 1, pages 346–348. IEEE Signal Processing Society, IEEE; New York, NY, May 1996.

[12] M. Finke, J. Fritsch, P. Geutner, K. Ries, and A. Waibel. The JanusRTk Switchboard/Callhome 1997 Evaluation System. In *Proceedings of the LVCSR Hub5-e Workshop*, May 1997.

[13] M. Finke and I. Rogina. Wide Context Acoustic Modeling in Read vs. Spontaneous Speech. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1743–1746. IEEE Signal Processing Society, IEEE Comput. Society Press; Los Alamitos, CA, April 1997.

[14] J. Fritsch and M. Finke. ACID/HNN: Clustering Hierarchies of Neural Networks for Context-Dependent Connectionist Acoustic Modeling. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 505–508. IEEE Signal Processing Society, IEEE; New York, NY, May 1998.

[15] J. Fritsch and A. Waibel. Hierarchies of Neural Networks for Connectionist Speech Recognition. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN'98)*, April 1998.

[16] R. Gross. Run-on Recognition in an On-line Handwriting Recognition System. Master's thesis, Universität Karlsruhe, June 1997.

[17] R. Haeb-Umbach and Ney H. Improvements in Time Synchronous Beam Search for 10000-Word Continuous Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 2:353–356, 1994.

[18] W. Herrmann. *Lehrbuch der modernen koreanischen Sprache.* Helmut Buske Verlag, 1994.

[19] H.-W. Hon and K.-F. Lee. CMU Robust Vocabulary-Independent Speech Recognition System. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 889–892. IEEE, IEEE; New York, NY, May 1991.

[20] X.D. Huang and M.A. Jack. Semi-continuous hidden Markov models for speech signals. *Computer, Speech and Language*, 3(3):239–251, july 1989.

[21] M.-Y. Hwang and X. Huang. Subphonetic modeling with Markov states – Senone. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 33–36. IEEE, IEEE; New York, NY, March 1992.

[22] F. Jelinek. A Fast Sequential Decoding Algorithm Using a Stack. *IBM Journal of Research and Development*, 13(6):675–685, November 1969.

[23] F. Jelinek. *Statistical Methods for Speech Recognition.* MIT Press, 1997.

[24] F. Jelinek, R.L. Mercer, and S. Roukos. *Advances in Speech Signal Processing*, chapter Principles of Lexical Language Modeling for Speech Recognition, pages 651–700. M. Dekker Publishers, New York, NY, 1991.

[25] T. Kemp, P. Geutner, M. Schmidt, B. Tornax, M. Weber, M. Westphal, and A. Waibel. The Interactive Systems Labs View4You Video Indexing System. In R.H. Mannell and J. Robert-Ribes, editors, *Proceedings of the International Conference on Spoken Language Processing, ICSLP 98*, volume 4, pages 1639–1942. Australian Speech Science and Technology Association, Incorporated, November 1998.

[26] P. Kenny. A* Admissible Heuristics for Rapid Lexical Access. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 682–692. IEEE, IEEE Comput. Soc. Press, Los Alamitos, CA, May 1991.

[27] D. Kiecza, T. Schultz, and A. Waibel. Data-Driven Determination of Appropriate Dictionary Units for Korean LVCSR. In *Proceedings of the*

International Conference on Speech Processing (ICSP'99), pages 323–327. Acoustical Society of Korea, August 1999.

[28] R. Kneser and H. Ney. Improved Backing-Off for M-Gram Language Modeling. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 181–184. IEEE Signal Processing Society, IEEE; New York, NY, May 1995.

[29] O.-W. Kwon, K. Hwang, and J. Park. Korean Large Vocabulary Continuous Speech Recognition Using Pseudomorpheme Units. In G. Prószéky, G. Németh, and J. Mándli, editors, *Proceedings of the Eurospeech 1999*, volume 1, pages 483–486. Scientific Society for Telecommunications, September 1999.

[30] L. Lamport. *Latex - A Document Preparation System.* Addison Wesley, second edition, 1994.

[31] A. Lavie, A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavalda, T. Zeppenfeld, and Z. Puming. Janus III: Speech-to-Speech Translation in Multiple Languages. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 99–102. IEEE, IEEE Comput. Soc. Press, Los Alamitos, CA, April 1997.

[32] A. Lavie, A. Waibel, L. Levin, D. Gates, M. Gavalda, T. Zeppenfeld, P. Zhan, and O. Glickman. Translation of Conversational Speech With Janus-II. In *Proceedings of the International Conference on Spoken Language Processing, ICSLP 96*, volume 4, pages 2375–2378. IEEE, IEEE, New York, NY, October 1996.

[33] H.-S. Lee, J. Park, and H.-R. Kim. An Implementation of Korean Spontaneous Speech Recognition System. In *Proceedings of the International Conference on Signal Processing Applications and Technology, ICSPAT 1996*, volume 2, pages 1801–1805, October 1996.

[34] K.-F. Lee. *Large-Vocabulary Speaker-Independent Continuous Speech Recognition: the SPHINX System.* PhD thesis, School of Computer Science, Carnegie Mellon University, May 1988.

[35] K.-F. Lee, S. Hayamizu, H.-W. Hon, C. Huang, J. Swartz, and R. Weide. Allophone Clustering for Continuous Speech Recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal*

Processing, volume 2, pages 749–752. IEEE, IEEE; New York, NY, April 1990.

[36] C.J. Leggetter and P.C. Woodland. Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer, Speech and Language*, 9(2):171–185, April 1995.

[37] H. Ney, L. Welling, S. Ortmanns, K. Beulen, and F. Wessel. The RWTH Large Vocabulary Continuous Speech Recognition System. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 853–856. IEEE Signal Processing Society, IEEE; New York, NY, May 1998.

[38] H.J. Nock, M.J.F. Gales, and S.J. Young. A Comparative Study of Methods for Phonetic Decision-Tree State Clustering. In *Proceedings of the Eurospeech 97*, volume 1, pages 111–114. The University of Patras, Wire Communications Laboratory, September 1997.

[39] F.Y.T. Park. *Speaking Korean, Book I.* 1977.

[40] D.B. Paul. Algorithms for an Optimal A* Search and Linearizing the Search in the Stack Decoder. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 693–696. IEEE, IEEE Comput. Soc. Press, Los Alamitos, CA, May 1991.

[41] D.B. Paul. The Lincoln Tied Mixture HMM Continuous Speech Recogniser. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 329–332. IEEE, IEEE Comput. Soc. Press, Los Alamitos, CA, May 1991.

[42] L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, volume 77 of *2*, pages 257–286. IEEE, feb 1989.

[43] M. Ravishankar. *Efficient Algorithms for Speech Recognition.* PhD thesis, School of Computer Science, Carnegie Mellon University, May 1996.

[44] I. Rogina. *Parameterraumoptimierung für Diktiersysteme mit unbeschränktem Vokabular.* PhD thesis, Universität Karlsruhe, June 1997.

[45] R.W. Schafer and L.R. Rabiner. *Digital Representations of Speech Signals*, pages 49–64. In Waibel and Lee [56], 1990.

[46] T. Schultz et al. Language Independent and Language Adaptive Large Vocabulary Speech Recognition. In R.H. Mannell and J. Robert-Ribes, editors, *Proceedings of the International Conference on Spoken Language Processing, ICSLP 98*, volume 5, pages 1819–1822. Australian Speech Science and Technology Association, Incorporated, November 1998.

[47] T. Schultz and A. Waibel. Das Projekt GlobalPhone: Multilinguale Spracherkennung. In B. Schroder, W. Lenders, W. Hess, and T. Portele, editors, *Proceedings of the 4th Conference on Natural Language Processing-KONVENS-98. Computers, Linguistics, and Phonetics between Language and Speech*, pages 179–189. Peter Lang, Frankfurt am Main, Germany, October 1998.

[48] T. Schultz and A. Waibel. Development of Multilingual Acoustic Models in the GlobalPhone Project. In *Proceedings of the first Workshop on Text, Speech, and Dialogue (TSD)*, pages 311–316. Masaryk University, September 1998.

[49] T. Schultz and A. Waibel. Multilingual and Crosslingual Speech Recognition. In *Proceedings of the DARPA Broadcast News Workshop*, pages 259–262. IEEE, February 1998.

[50] T. Schultz and A. Waibel. Experiments towards a Multi-language LVCSR Interface. In *Second International Conference on Multimodal Interfaces (ICMI '99)*, January 1999.

[51] T. Schultz and A. Waibel. Language Adaptive LVCSR Through Polyphone Decision Tree Specialization. In *Proceedings of the Workshop on Multi-lingual Interoperability in Speech Technology (MIST '99)*, September 1999.

[52] T. Schultz, M. Westphal, and A. Waibel. The GlobalPhone Project: Multilingual LVCSR with JANUS-3. In *Multilingual Information Retrieval Dialogs: 2nd SQEL Workshop*, pages 20–27, April 1997.

[53] S.-C. Song. *201 Korean Verbs - fully conjugated in all the forms.* Barron's Educational Series, Inc., 1988.

[54] S. Umesh, L. Cohen, and D. Nelson. Frequency-Warping and Speaker-Normalization. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 983–986. IEEE Signal Processing Society, IEEE Comput. Society Press; Los Alamitos, CA, April 1997.

[55] A. Waibel, M. Bett, M. Finke, and R. Stiefelhagen. Meeting Browser: Tracking and Summarising Meetings. In *Proceedings of the DARPA Broadcast News Workshop*, 1998.

[56] A. Waibel and K.-F. Lee, editors. *Readings in Speech Recognition*. Morgan Kaufmann, 1990.

[57] M. Westphal. Dimensionalitätsreduktion von Sprachsignalen mit statistischen und neuronalen Methoden. Master's thesis, Universität Karlsruhe, January 1994.

[58] M. Westphal, T. Schultz, and A. Waibel. Linear Discriminant – A New Criterion for Speaker Normalization. In R.H. Mannell and J. Robert-Ribes, editors, *Proceedings of the International Conference on Spoken Language Processing, ICSLP 98*, volume 3, pages 827–830. Australian Speech Science and Technology Association, Incorporated, November 1998.

[59] M. Westphal and A. Waibel. Towards Spontaneous Speech Recognition for On-Board Car Navigation and Information Systems. In G. Prószéky, G. Németh, and J. Mándli, editors, *Proceedings of the Eurospeech 1999*, volume 5, pages 1955–1958. Scientific Society for Telecommunications, September 1999.

[60] P.C. Woodland, C.J. Leggetter, J.J. Odell, V. Valtchev, and S.J. Young. The 1994 HTK Large Vocabulary Speech Recognition System. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 73–76. IEEE Signal Processing Society, IEEE; New York, NY, May 1995.

[61] M. Woszczyna. *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition*. PhD thesis, Universität Karlsruhe, February 1998.

[62] M. Woszczyna and M. Finke. Minimizing Search Errors due to Delayed Bigrams in Real-Time Speech Recognition Systems. In *Proceedings of the*

International Conference on Acoustics, Speech and Signal Processing, volume 1, pages 137–140. IEEE, IEEE; New York, NY, May 1996.

[63] M. Woszczyna, M. Finke, T. Kemp, A. McNair, A. Lavie, L. Mayfield, M. Maier, I. Rogina, T. Sloboda, A. Waibel, P. Zahn, and T. Zeppenfeld. Janus II - Translation of Spontaneous Conversational Speech. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 409–412. IEEE, IEEE; New York, NY, May 1996.

[64] S.J. Young. The General Use of Tying in Phoneme-Based HMM Speech Recognisers. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 569–572. IEEE, IEEE; New York, NY, March 1992.

[65] S.J. Young. A Review of Large Vocabulary Continuous Speech. *IEEE Signal Processing Magazine*, 13(5):45–57, September 1996.

[66] H. Yu, M. Finke, and A. Waibel. Progress in Automatic Meeting Transcription. In G. Prószéky, G. Németh, and J. Mándli, editors, *Proceedings of the Eurospeech 1999*, volume 2, pages 695–698. Scientific Society for Telecommunications, September 1999.

[67] T. Zeppenfeld, M. Finke, K. Ries, M. Westphal, and A. Waibel. Recognition of Conversational Telephone Speech Using the Janus Speech Engine. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1815–1818. IEEE, IEEE Comput. Soc. Press; Los Alamitos, CA, April 1997.

[68] P. Zhan, K. Ries, M. Gavalda, D. Gates, A. Lavie, and A. Waibel. Janus-II: Towards Spontaneous Spanish Speech Recognition. In *Proceedings of the International Conference on Spoken Language Processing, ICSLP 96*, volume 4, pages 2285–2288. IEEE, IEEE, New York, NY, October 1996.

[69] P. Zhan and A. Waibel. Vocal Tract Length Normalization for Large Vocabulary Continuous Speech Recognition. Technical Report CMU-CMT-97-150, Carnegie Mellon University, Center for Machine Translation, May 1997.

[70] P. Zhan and M. Westphal. Speaker Normalization Based on Frequency Warping. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 1039–1042. IEEE Signal Processing Society, IEEE Comput. Society Press; Los Alamitos, CA, 1997 April.

[71] P. Zhan, M. Westphal, M. Finke, and A. Waibel. Speaker Normalization and Speaker Adaptation – A Combination for Conversational Speech Recognition. In *Proceedings of the Eurospeech 97*, volume 4, pages 2087–2090. The University of Patras, Wire Communications Laboratory, September 1997.

[72] World Wide Web:

```
http://isl.ira.uka.de/~tanja/gp/index.html
http://www.unicode.org/
http://www.chosun.com/w21data/html/news/
ftp://ftp.kaist.ac.kr/hangul/code/hcode/
http://pantheon.yale.edu/~jshin/faq/qa8.html:
   Contains a description of the hangul
   code translation tool hcode.
```